Tampere University

SYEDA SAKIRA HASSAN

Regularization in Machine Learning with Applications in Biology

Tampere University Dissertations 59

Tampere University Dissertations 59

SYEDA SAKIRA HASSAN

Regularization in Machine Learning with Applications in Biology

ACADEMIC DISSERTATION
To be presented, with the permission of
the Faculty Council of the Faculty of Information Technology
and Communication Sciences of Tampere University,
for public discussion in the auditorium TB109
of the Tietotalo, Korkeakoulunkatu 1, Tampere,
on 17 May 2019, at 12 o'clock.

ACADEMIC DISSERTATION
Tampere University, Faculty of Information Technology and Communication Sciences
Finland

Responsible supervisor and Custos	Associate Professor Heikki Huttunen Tampere University Finland	
Supervisors	Professor Olli Yli-Harja Tampere University Finland	Assistant Professor Ville Santala Tampere University Finland
Pre-examiners	Associate Professor Ulisses M. Braga-Neto Texas A&M University United States	Associate Professor Tapio Pahikkala University of Turku Finland
Opponents	Professor Pekka Neittaanmäki University of Jyväskylä Finland	Associate Professor Ulisses M. Braga-Neto Texas A&M University United States

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Cover design: Roihu Inc.

ISBN 978-952-03-1084-4 (print)
ISBN 978-952-03-1085-1 (pdf)
ISSN 2489-9860 (print)
ISSN 2490-0028 (pdf)
http://urn.fi/URN:ISBN:978-952-03-1085-1

PunaMusta Oy – Yliopistopaino
Tampere 2019

Abstract

Over recent years, data-intensive science has been playing an increasingly essential role in biological discovery and biomedical science. The explosion of information in biology poses challenges in organizing data, discovering relevant information from the data, extracting salient features, and providing a comprehensive understanding of the overall biological process. Traditional manual approaches are no longer a feasible solution due to the heterogeneous, complex and unstructured nature of the biological data. Therefore, a framework for efficient, robust, automated, and fast data-intensive methods and pipelines are required to handle and explore these data.

The field of machine learning provides a comprehensive array of computational tools for analyzing such data. The aim of this thesis is to learn a simple model from such biological data, where the learned model illustrates an overview of the underlying data-generating process. This, in turn, should allow the extraction of salient features for predictive analysis of unobserved data. In this thesis, we consider solving the biological problem from the context of the dimension of the data. High-dimension data refers to the phenomenon where the number of data points is larger than the features describing the data points, or vice versa, or both. High-dimensionality can lead to the risk of overfitting of the model, a phenomenon in which the performance of the model is poorly described for the predictive analysis of unobserved data.

To this end, an attempt to add additional information or to modify the learning algorithm, a strategy known as regularization, is indispensable to increase the generalization capability of the model. The results of this study indicate that a regularized version of simple linear models often outperforms more sophisticated methods. Moreover, implicit automated feature-selection capabilities in sparse regularized parameter estimations have made a significant contribution to the thesis. We also utilize a powerful ensemble tree-based method, random forest, which is effective for discovering nonlinear relationships among features as well as providing feature ranking.

Another important aspect in the learning process considered in this thesis is model selection i.e., the selection of one model from a hypothesized set of possible models. The set of candidate models is constructed by setting different values for the models' hyperparameters before initializing the learning process. It is shown that an alternative Bayesian approach is computationally faster and has lower error rates than the traditional approaches to model selection, such as grid search and cross-validation. Moreover, we propose a closed-form expression for the area under the receiver operating characteristic curve, a performance metric, in the context of a linear classifier.

In addition, we consider the unsupervised machine-learning paradigm in which the ground truth of biological data is not provided, such as for microarray gene expression profiling. The results show that clustering methods can be used effectively to explore the data and

discover similarities. Our results indicate that careful selection of the machine-learning approaches can create powerful, yet simple computational modeling and analysis that can provide new and useful insights into heterogeneous biological applications.

Preface

This study was carried out at the Laboratory of Signal Processing and the Laboratory of Chemistry, Tampere University, formerly Tampere University of Technology (TUT), during 2014 – 2018. The funding was provided by Presidential Graduate School in Tampere University of Technology.

I would like to express my sincere gratitude to my supervisors, Associate Professor Heikki Huttunen, Professor Olli Yli-Harja and Assistant Professor Ville Santala. I am grateful to Assoc. Prof. Heikki Huttunen, the head of Machine Learning Group (MLG) for his excellent guidance, unparalleled support, and encouragement that let me all the way through this study. I would like to thank my co-supervisor, Prof. Olli Yli-Harja for giving me the opportunity to conduct my research in the field of signal processing. I would also like to extend my gratitude towards Professor Matti Karp and Assistant Prof. Ville Santala for providing me the opportunity to work in the collaborative environment in the Laboratory of Chemistry and Bioengineering.

Assoc. Prof. Ulisses Braga-Neto and Assoc. Prof. Tapio Pahikkala are kindly acknowledged for reviewing this thesis. I would also like to extend my gratitude to Prof. Pekka Neittaanmäki and Assoc. Prof. Ulisses Braga-Neto for agreeing to act as my opponents.

I am highly indebted to Dr. Tommi Aho for his excellent guidance and support during my research. Without his continuous support, guidance and advice it would not have been possible to lay the foundations for my research career. I am very grateful to all my colleagues and co-authors for their help and support throughout my studies. Especially, I would like to thank Assoc. Prof. Jussi Tohka, University Lecturer Pekka Ruusuvuori, Dr. Muhammad Farhan, Dr. Rahul Mangayil, Dr. Reija Autio and Jari Niemi, MSc., for their excellent advice and comments in various stages of this research. I would like to thank Jukka, Bishwo, Francesco, Pedram, Yue, and Yi for providing such an enjoyable and productive work environment. I would also like to thank Ms. Ulla Siltaloppi, Ms. Elina Orava, Ms. Virve Larmila, Ms. Pirkko Ruotsalainen, Ms. Päivi Salo and other staff at the Laboratory of Signal Processing for helping me with all the practical arrangements.

Last but not least, I am grateful to my parents, sisters, and brother for their tremendous and continuous support throughout my life. Finally and above all, I am thankful to my husband Dr. Sharif Chowdhury for his tremendous inspiration and immense support all the time and to my daughter, Elina, whose smile makes me believe in myself. I dedicate this thesis to my father, Hassan, who wanted to see the realization of this thesis very much.

Sakira Hassan
Klaukkala, Nurmijärvi
April 29, 2019

Contents

Acronyms

ACC	Accuracy
AML	Acute myeloid leukemia
ANN	Artificial neural network
AUROC	Area under the receiver operating characteristic curve
BEE	Bayesian minimum mean square error estimator
BLUE	Best linear unbiased estimator
CART	Classification and regression trees
CBAUROC	Closed-form Bayesian AUROC
c-di-GMP	Cyclic di-guanosine monophosphate
CIFAR	Canadian institute for advanced research
CD	Cluster of differentiation
CV	Cross-validation
DoE	Design-of-experiments
DNA	Deoxyribonucleic acid
DNN	Deep neural network
DP	Degree of polymerization
DREAM	Dialogue for reverse engineering assessment and methods
E	Error rate
EBAUROC	Empirical Bayesian AUROC
FlowCAP	Flow cytometry: Critical assessment of population identification methods
FN	False negative
FP	False positive
FPR	False positive rate
FSC	Forward scatter
GLM	Generalized linear model
H_2	Hydrogen
HMM	Hidden Markov model
ICA	Independent component analysis
IRLS	Iteratively reweighted least squares
LASSO	Least absolute shrinkage and selection operator
LDA	Linear discriminant analysis

LOO	Leave-one-out
LOOCV	Leave-one-out cross-validation
LR	Logistic regression
MAE	Mean absolute error
MEG	Magnetoencephalography
MLR	Multiple linear regression
MLE	Maximum likelihood estimation
MMSE	Minimum mean squared estimator
MSE	Mean squared error
PAT	Process analytical technology
PCA	Principal component analysis
QbD	Quality by design
R^2	Coefficient of determination (R-squared)
RF-ACE	Random forests with artificial contrast ensembles
RFE	Recursive Feature Elimination
RNA	Ribonucleic acid
mRNA	messenger RNA
ROC	Receiver operating characteristic curve
RSS	Residual sum of squares
SSC	Side scatter
SOM	Self-organizing map
SVM	Support vector machines
TN	True negative
TP	True positive
TPR	True positive rate

Mathematical notations

x_i, y	Scalars
s	A constant scalar
$\mathbf{x}, \mathbf{y}$	Vectors
$\mathbf{x}^c, y_c$	Observations for class c
$\mathbf{X}$	Matrix
$\mathbf{x}^T$	Transpose of $\mathbf{x}$
p	Dimensionality or the number of features
n	The number of observations
c	A discrete number of classes
$\mathbb{R}$	The set of real numbers
$\mathbb{R}^p$	The set of p-dimensional real numbers
$f(\cdot), J(\cdot)$	Functions
$g_i(\cdot)$	A discriminant function of class c_i
$L(\cdot, \cdot)$	The loss function
$\mathbb{F}$	Feature space
$\mathcal{R}_i$	The i-th decision region
$\lvert \cdot \rvert$	The absolute function or Laplace function
$\exp(\cdot)$	The exponential function
$\log(\cdot)$	The natural logarithm function
$\text{sgn}(\cdot)$	The sign function
$\text{trace}(\cdot)$	Matrix trace
$\arg\min_{\boldsymbol{\theta}}\{\cdot\}$	Minimizing argument $\boldsymbol{\theta}$
$\eta(\cdot)$	A logistic function
$\det(\mathbf{X})$	The determinant of a matrix $\mathbf{X}$
$\hat{f}$	Function approximation of $f(.)$
$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}_{ridge}, \hat{\boldsymbol{\beta}}$	Estimates of model parameters
$\hat{y}$	Estimate of y
$\boldsymbol{\mu}_c$	Mean vector of p-dimension for class c
$\boldsymbol{\Sigma}_c$	Covariance matrix of $p \times p$-dimension for class c
$X \triangleq Y$	X is defined to be logically equivalent to Y
$\mathbb{E}[x]$	Expected value of random variable x
ε	True classification error

$\varepsilon_c(\boldsymbol{\Theta}_c)$	True classification error for class c
$\mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma})$	Multivariate Gaussian distribution with mean $\mathbf{m}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{W}^{-1}(\mathbf{S}, \kappa)$	Inverse Wishart distribution with $p \times p$ matrix $\mathbf{S}$ and the number of degrees of freedom κ
$\Gamma^{-}1(\alpha, \beta)$	Inverse Gamma distribution with parameters α and β
$\nu, \kappa, \mathbf{S}, \mathbf{m}$	Prior hyperparameters of the Bayesian model
$\nu^*, \kappa^*, \mathbf{S}^*, \mathbf{m}^*$	Posterior hyperparameters of the Bayesian model
$\Phi(\cdot)$	The unit normal Gaussian cumulative distribution function
$I(x; \alpha, \beta)$	An incomplete beta function with parameter α and β
$\mathbf{a}_c, b_c, \mathbf{z}, A, \alpha, \beta$	Variables
σ, σ^2	Standard deviation, variance
n_c	Number of observations for class c
$\nu_c^*, \mathbf{m}_c^*$	Posterior hyperparameters for class c
$\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\Sigma}}_c$	Mean and covariance matrix for observations of class c
$\boldsymbol{\Theta}_c, \boldsymbol{\Theta}_c^*$	Prior and posterior distribution parameters of class c
$\mathbf{U}$	A unitary matrix
$\mathbf{m}_1, \mathbf{m}_2$	Vectors of means of dimension p for class c_1 and class c_2
$\mathbf{S}_W$	Pooled within-class sample covariance matrix of $p \times p$ dimensions
$\mathfrak{D}_{\texttt{train}}$	A set containing training data
$\mathfrak{D}_{\texttt{train}}{}'$	Subsets of $\mathfrak{D}_{\texttt{train}}$
β_i	The i-th model parameter or coefficient
$\boldsymbol{\beta}$	The vector of model parameters or coefficients excluding β_0
$\boldsymbol{\theta}$	The vector of model parameters or coefficients including β_0
T	Threshold
λ	A model hyperparameter
$\Omega(\cdot)$	A regularization function
$\psi(.)$	An activation function or link function
γ	A mixing hyperparameter
$\mathbf{I}$	A $p \times p$ identity matrix
$\Vert \cdot \Vert_{\ell_1}$	Vector ℓ_1 norm
$\Vert . \Vert$, $\Vert \cdot \Vert_{\ell_2}$	Vector ℓ_2 norm
$\Vert \cdot \Vert_{\ell_q}$	Vector ℓ_q norm
$\mathcal{O}(\cdot)$	The big-O notation for algorithm complexity
$\mathbb{P}\mathrm{r}(\cdot)$	Prior probability
$\mathbb{P}\mathrm{r}(\cdot, \cdot)$	Conditional probability or likelihood function
$\mathbb{P}\mathrm{r}^*(\cdot)$	Posterior probability

List of Publications

This thesis is a compilation of the following publications, which are referred to in the text as **Publication I, Publication II, Publication III, Publication IV** and **Publication V**.

I Syeda Sakira Hassan, Muhammad Farhan, Rahul Mangayil, Heikki Huttunen, and Tommi Aho. "Bioprocess data mining using regularized regression and random forests," *BMC Systems Biology*, vol 7, no. Supp 1, pp. S(5), Aug. 2013.

II Sara Urmersbach, Tommi Aho, Thomas Alter, Syeda Sakira Hassan, Reija Autio, and Stephan Huehn. "Changes in global gene expression of Vibrio parahaemolyticus induced by cold-and heat-stress," *BMC microbiology*, vol 15, no. 1, pp. 229 Oct. 2015.

III Syeda Sakira Hassan, Pekka Ruusuvuori, Leena Latonen, and Heikki Huttunen. "Flow Cytometry-Based Classification in Cancer Research: A View on Feature Selection," *Cancer informatics*, vol 14, no. S(5), pp. 75-85, Feb. 2016.

IV Syeda Sakira Hassan, Rahul Mangayil, Tommi Aho, Olli Yli-Harja, and Matti Karp. "Identification of feasible pathway information for c-di-GMP binding proteins in cellulose production," *Joint Conference of the European Medical and Biological Engineering Conference (EMBEC) and the Nordic-Baltic Conference on Biomedical Engineering and Medical Physics (NBC)*, International Federation for Medical and Biological Engineering (IFMBE) Proceedings, vol 65, pp. 667-670, Jun. 2017.

V Syeda Sakira Hassan, Jari A. Niemi, Jussi Tohka, and Heikki Huttunen, "Bayesian Receiver Operating Characteristic Metric for Linear Classifiers," *Pattern Recognition Letters, in review.*

1 Introduction

Today, we live in a *data deluge* era which has lead us to the need for data-intensive science to deal with the explosion of information in biology, physics, social media, security and e-commerce. The leaps in digital data acquisition (such as digital cameras, medical scanners), digital communications (such as Wi-Fi, 4G), and digital signal processing technologies (such as audio, video, speech and image processing and robotics) have accelerated the accumulation in the volume of data to thousands-fold. For example, Twitter, created in 2006, generates around 500 million textual data everyday [1]. Users of Facebook, a social networking website, upload 300 million photos a day, on average [2]. The existing trend in the growth of digital data is doubling each year [3].

This surge in digital data is also dragging the study of living organisms towards an exponential growth in the amount of biological data available. Researchers have been benefited from the discovery of the structure of deoxyribonucleic acid (DNA) [4] and finding gene sequences in it. Different genome projects, such as the Human Genome Project, 1000Genomes, ENCODE and HapMap are generating a diverse yet vast amount of omics data even within limited lab settings [5]. For instance, in 2008, GenBank, a public database for the nucleotide sequences, deposited the nucleotide sequences of more than $260,000$ organisms [6]. The amount of sequence data has doubled every seven months over the last decade [7].

1.1 Motivation for this thesis

Besides just collecting and storing these huge quantities of digital data from various domains, it is also necessary to transform these data into knowledge. With this objective, several methods and tools have been developed in recent years for mapping raw data into compact representations, providing an overview of the underlying data-generating process, and enabling predictive analysis to estimate the future value of any unobserved data [8]. The traditional statistical and machine-learning approaches, such as hypothesis testing, analysis of variance, linear regression, and maximum likelihood estimation have been designed to cope with such data, but are limited by the computing memory available [9, 10]. Due to this limited capacity, we used to measure data with many observations (such as the number of persons) and a few effectively chosen features for each observation (such as the blood pressure, weight and height of an individual).

Today, the explosion of information has not only shifted toward ever more observations, but the number of features covered, known as the *dimension* of the data, has also increased immensely. For instance, one single measurement from a microarray can yield from thousands to tens of thousands of gene expression levels. These measurements are expensive, so the number of observations (say, n) are in two to three digits, but the

dimensions of the data (say, p) is much larger, i.e., $p \gg n$ [11]. A good example is the analysis of flow cytometry data in cancer research, where we measure different features of an individual cell, such as its granularity, size, and other chemical properties of the cell [12]. A flow cytometer can process from a thousand to tens of thousands of cell in a single session and is thus able to produce large quantities of measurement data, even though the observations are limited to only a few patients.

The aforementioned examples are common scenarios for high-dimensional problems. Such data poses challenges for many traditional methods of statistical analysis, particularly in terms of variance, overfitting and generalization, which are a major concern in this context. The analysis of high-dimensional data, therefore, requires either modification to the approaches designed for the $p < n$ scenario, or completely new approaches [13]. As a result, dimension-reduction approaches and regularized approaches are often the methods of choice. The benefit of regularization has long been recognized in machine learning in the form of the modeling methods, such as ridge regression [14], lasso [15] and support vector machines [16]. The impact of these regularized approaches is now visible in many application areas, including biological science. Some of the recent applications in regularization include genetic risk prediction [17], a vivo imaging method for early cancer detection [18] and gap-filling approaches for limited-angle x-ray imaging techniques [19].

In addition to the above, a significant part of omics data is heterogeneous and unstructured, which also poses challenges in extracting salient features and statistical relationships from the data in order to construct a meaningful representation of the relevant information. Managing raw biological data, extracting information from published literature, annotating and translating the data for clinical and research use still remains an elusive goal for several reasons. Even though an individual omics domain presents distinct and important information, no single omics domain is sufficient to provide a comprehensive understanding of complex human biology.

1.2 The research questions and objectives of this thesis

The increasing demand for data-intensive frameworks for biological discoveries and biomedical science has raised the necessity for efficient, robust, automated and fast pipelines. Machine learning, on the other hand, has given rise to a large pool of data analysis tools and methodologies for developing sophisticated, reliable and fast classification or regression models. These models are used for analyzing high-dimensional data and can learn complex and subtle patterns hidden underneath the data. The primary objective of this thesis is to study the complex and heterogeneous data in various biological systems and to discover patterns of interest. To this end, this study poses the following research questions:

Q1 What kind of machine-learning pipelines are suitable for analyzing complex, heterogeneous and unstructured biological data?

Q2 What are the most significant challenges in evaluating the accuracy of machine-learning models and comparing them against each other?

The first research question assumes a framework for suitable machine-learning approaches to analyze such challenging biological data and to answer the biological problem in question. The results of this research are reported in **Publications I**, **II**, **III** and **IV**. **Publications III** and **V** address the second research question. In particular, we

study the data analysis with a limited number of observations relative to the number of features. Moreover, we concentrate on the accuracy and the stability of feature-selection approaches in the context of machine-learning pipelines. We also focus on gene-expression data analysis. In addition to these, we introduce an accuracy metric for assessing the accuracy of a model without the need for repetitive exposure of the available data.

In summary, the objectives of this thesis are as follows:

- To develop models in order to get an overview of the data-generating process in biological and biomedical applications using regularization approaches.
- To identify significant features in these applications using feature-selection approaches.
- To ensure the accuracy and the stability of feature-selection approaches.
- To develop an accuracy metric to quantify the accuracy of the classifier without the need for computationally intensive estimators.
- To develop robust and efficient data analysis pipelines for high-throughput biological data using unsupervised learning.

1.3 Thesis outline

This thesis is organized as follows: Chapter 2 provides an overview of the characteristics of machine learning in biological data and some common challenges associated with high-dimensional data. Chapter 3 provides the theoretical background to machine-learning approaches, which leads towards finding the answer to the research question, **Q1**. Chapter 4 provides an overview of the assessment criteria and introduces a novel criterion. This leads to plausible solutions for the research question, **Q2**. The case studies and key findings are presented in Chapter 5. Chapter 6 summarizes the publications and the author's contribution. Finally, Chapter 7 provides some concluding remarks.

2 Characteristics of machine learning in biology

The interconnection between the field of machine learning and biology has a long and complex history. One of the earliest mathematical models, the *perceptron* [20], is the precursor to the modern artificial neural network (ANN), which imitates the behavior of a neuron in the human brain. Later, deep neural networks (DNN) emerged from this field. Another early work linking these two paradigms dates back to the 1990*s*. Krogh et al. effectively explored the Hidden Markov Model (HMM), a statistical machine-learning tool in protein modeling [21] and gene prediction [22]. The model gained a lot of attention in the computational biology community and it has been applied to various problems in computational biology ever since.

Most of these applications can be characterized in a machine-learning workflow that involves the following stages: i) data preprocessing and feature selection or extraction; ii) model training; iii) model evaluation and selection, and iv) prediction. Figure 2.1 gives us an intuitive understanding of the data flow and the interconnection between these stages.

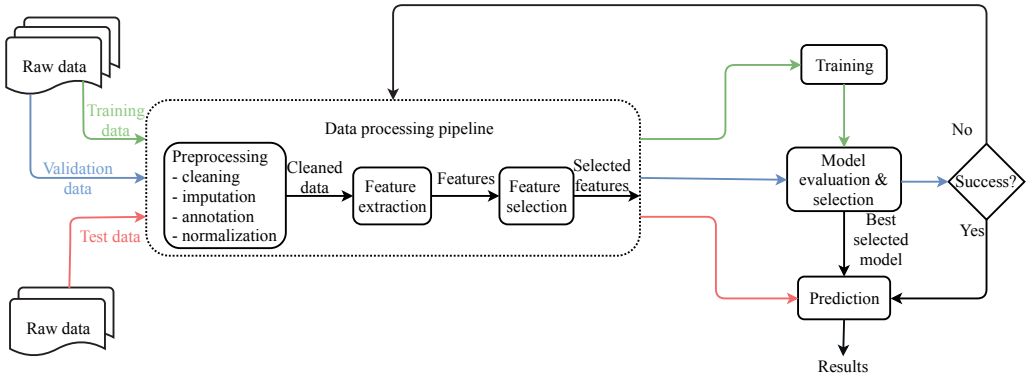


Figure 2.1: The flow of data in the machine-learning paradigm. A machine-learning system can be designed for a problem specific to an application domain using one or more of the stages shown here. Raw data are collected for both training and testing the system. The data processing stage extracts useful features from the raw data. In the training stage, some or all of the data are used to determine parameters for the model. The evaluation stage may require the repetition of various stages to obtain the desired results.

The different stages in the machine-learning data flow are described below:

- **Data collection.** Data are collected from various sources, such as sensors, cameras,

speedometers, gene expression profiling, and so on. Raw data are usually unformatted and direct measurements from devices have been used for specific application areas.

- **Data processing.** This stage is required to improve the quality of the data and to ensure effective data analysis. The data processing is the most time-consuming stage in the machine-learning pipeline and may require about $70 - 80\%$ of the total time [23].
  - **Data preprocessing.** Various tools and methods are available to transform the raw data into a meaningful representation for intelligent data analysis. For example, imputation can be used to replace missing information with substituted values.
  - **Feature extraction.** The next step is to generate *features* from the cleaned data set. The features are the distinguishable properties of the data that can help to differentiate between the categories of input patterns. The idea is to represent or distill the data in such a way that it enables easy model training and facilitates the generalization of the model.
  - **Feature selection.** Feature selection is a critical stage in the data flow process. However, not all machine-learning pipelines include a feature-selection stage, e.g. neural networks. In the feature-selection procedure, a subset of the data is chosen, whereas in the feature-extraction process, the data is transformed into a set of features that are informative and nonredundant. The objective is to reduce the number of features in order to remove redundant and irrelevant features to ensure faster training. In some problems, there are no automatic feature-selection stages, so the process is performed by domain experts. On the other hand, many machine-learning algorithms embed the feature-selection process implicitly as part of the learning models.
- **Training.** The *models* are trained. More specifically, the training data are repeatedly presented to the model and a learning algorithm will adjust the model parameters to minimize any prediction error.
- **Evaluation and selection.** In this stage, the *best* or *final* model is selected from a set of competitive models based on performance criteria such as minimum mean squared error. The criteria are measured either with a validation set, or in some cases taken directly from the training data.
- **Prediction.** The accuracy of the final model is assessed using independent test data.

The primary goal of machine learning is to construct a generalized model which can perform well both for available data and for unseen data. This requires training the model with a data set that has all the possible combinations of patterns. However, the data is limited and it is difficult to ensure the quality of the learned model for unseen observations. A common strategy is to split the available data set into *training*, *validation*, and *test* sets as shown in Figure 2.2. The training set is used to train the model, i.e., to fit the parameters of the model. The validation set is used to evaluate the performance of the model, such as for different combinations of hyperparameter values. The test set is used to provide an unbiased evaluation of the selected model. In many situations, it is

difficult to obtain a separate independent test set and the available data set is divided into training and validation sets only. In this case, the validation set can be treated as a test set. However, this should be done with caution as repeated design choices based on the validation set's performance will lead to overlearning (discussed in Section 2.5) in the validation set as well. In this thesis, we use three different sets whenever possible.

Figure 2.2: Splitting a data set into 3 parts: training, validation, and test sets [13].

The construction of a good model depends on the characteristics of the data (such as the number of observations and the dimensionality) and the complexity of the learning algorithms (such as the hyperparameters). The rest of this chapter will discuss how the nature of data affects the design of the models.

2.1 Common data-related challenges in machine learning

Figure 2.3 illustrates a few of the challenges that often occur in a real-life data collection process. Many machine-learning algorithms may not perform correctly if some observations are *missing*. Missing data can occur for various reasons, such as erroneous data transmission, sensor failure, or improper data collection. For example, the respondents in a survey may be reluctant to respond to all the questionnaires. In these cases, missing observations can be removed or replaced with substituted values. An observation point that is distant from other observations is identified as an *outlier*. The value is too unlikely to be generated by the observation process. For example, a negative value of the heart rate measurement of a patient is definitely an outlier. Some of the possible sources of outliers are malfunctions or changes in the data-collection process, contamination in data transmission, and the natural deviation from a population.

Redundancy in data is the correlation between two or more features. If we have two features, for instance, "age" and "height" in a data set, we expect that there is a high correlation between these two features. The high degree of redundancy and irrelevant data can degrade the performance of machine-learning approaches. *Multicollinearity* is another phenomenon that occurs when two or more variables that describe the responses, are highly correlated. This creates a redundancy issue and the results can be skewed i.e., the model can be sensitive to changes in data. If a variable is an exact linear combination of others, a situation known as perfect multicollinearity, this can lead to a rank-deficiency issue. *Sparsity* in data occurs when there is an irregularity in the measurements or in the data-collection process and therefore, most of the observations have only zero values. This is a common scenario in a publicly available data set and can lead to an unsatisfactory model performance.

In classification problems, some of the existing methods assume that data are evenly distributed among classes. *Imbalanced* class distribution occurs when one group or class of data are insufficiently represented, i.e., observations belonging to one class are heavily outnumbered by observations belonging to the other class. The performance of most classification algorithms is accuracy driven, where the goal is to minimize the overall error and to maximize the classification accuracy. However, the classification accuracy tells us very little about the minority class in an imbalanced data set, and choosing accuracy as a

performance criterion can provide inaccurate and misleading information. Algorithms such as ordinary linear regression assume that all residuals (deviations from the true value) are drawn from a population that has a constant variance. This assumption is known as *homoscedasticity*. Nevertheless, in practice, this assumption is invalid. That is, the residuals are *heteroscedastic* and the variance of the residuals can vary across classes.

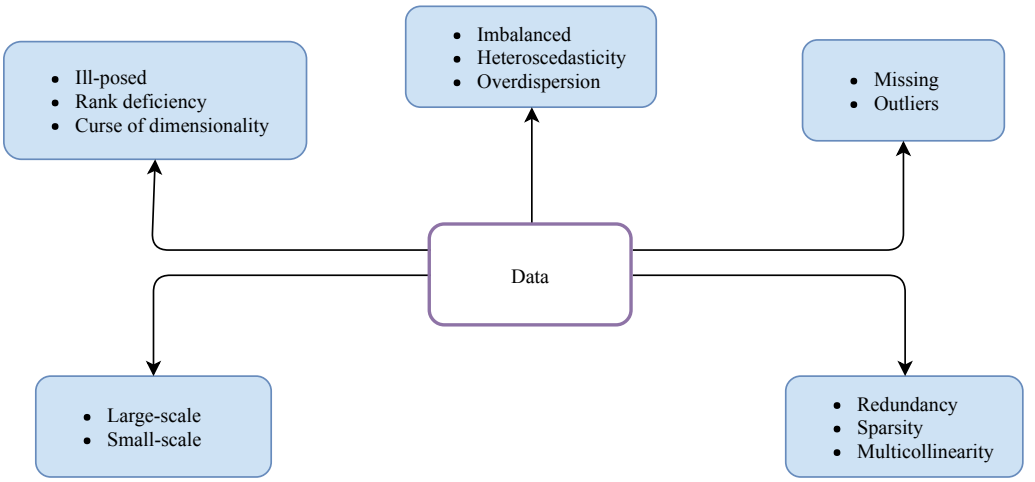


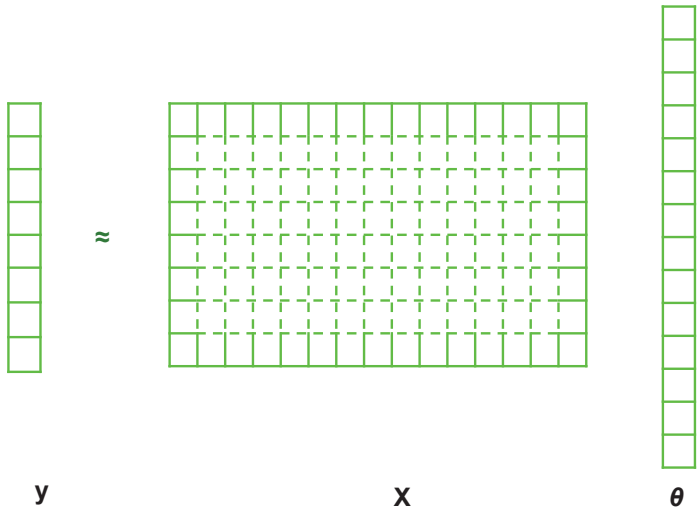
Figure 2.3: Common challenges associated with data.

The studies in medical domains are often limited to *small-scale* data due to their complexity and the high cost of collecting data from patients. Small-scale data are often difficult to handle as the learned model can easily be overfitted and affected by outliers. On the other hand, in a *large-scale* data set, the increase in the number of observations will drastically affect the time and memory needed to train a model. For example, the support vector machine (SVM) algorithm has a training time complexity of $\mathcal{O}(n^3)$ and a space complexity of $\mathcal{O}(n^2)$ [24], where n is the number of observations.

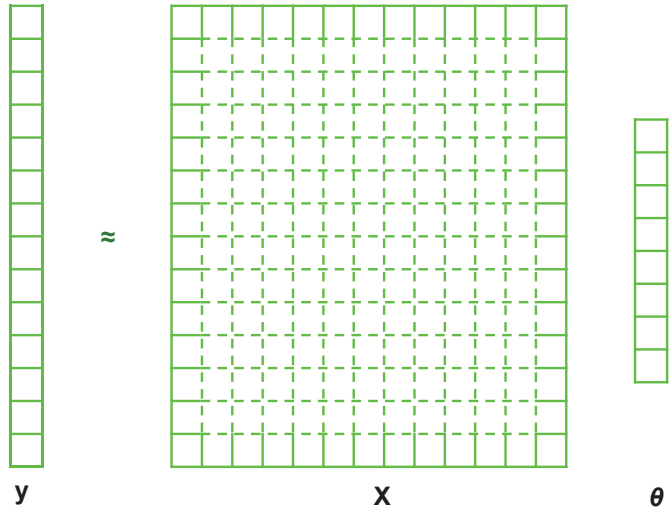
Rank deficiency is the lack of sufficient information in the data to estimate parameters uniquely. For example, we cannot estimate a unique quadratic model using only two data points. An *Ill-posed* situation occurs when small changes in the data lead to a large change in the solutions. The *curse of dimensionality* refers to a phenomenon encountered while dealing with a high-dimensional space. The dimension describes the number of features or attributes in the data set. For a fixed number of observations, the prediction accuracy and effectiveness of the algorithm decrease as the number of dimensions increases. This will be discussed further in Section 2.6.

2.2 Under-determined scenario

Imagine, a person purchased 20€-worth of apples and oranges from a grocery shop and we also received information that the person bought 2 apples and 2 oranges. With this information, we would be able to eliminate some possibilities, e.g. the unit price of each fruit cannot be more than 20€. However, we do not have enough information to find the actual unit prices of these two fruits. This situation is known as *under-determination*, where the information available is insufficient to correctly identify which conclusion to reach on the basis of that information. An under-determined system also occurs in



(a)



(b)

Figure 2.4: An overview of (a) an under-determined scenario, and (b) an over-determined scenario. Here, $y = \mathbf{X}\,\boldsymbol{\theta}$ represents a system of linear equations. In an under-determined system, $\mathbf{X}$ is a wide matrix (the number of rows are fewer than columns). On the other hand, an over-determined system has a tall matrix, $\mathbf{X}$ (the number of columns are fewer than rows).

signal processing, inverse problems and genomic data analysis, where the number of observations (say, n) is smaller than the number of features (say, p). This is equivalent to

an under-determined linear equation with fewer equations than unknowns, as illustrated in Figure 2.4 (a). An under-determined problem, in general, has an infinite number of solutions, if exists [25]. Nevertheless, regularization approaches are introduced in solving the under-determined problems [26, 27].

Let us consider the aforementioned example. We can infer the following information:

- In this example, the number of observations is $n = 1$ and the number of features is $p = 2$.
- We need to find the unit prices of the apple and the orange such that $\beta_1 \times 2 + \beta_2 \times 2 = 20€$, where β_1 and β_2 are the unit prices of the apples and oranges, respectively.
- We also know that $|\beta_1 + \beta_2| \leq 20$.
- Here, more than one solution is possible.

One of the solutions might be to regularize by posing a constraint to minimize the sum of the unit prices of the fruits. The constraint could be, for instance, the unit prices are integer values or the apple is cheaper than the orange. However, we can still have more than one solution.

Table 2.1: List of a few data sets in machine-learning research.

Data set name	Number of observations	Number of features	Task type	Reference
Arcene*	253	15,154	Classification	**Publication V**, [28]
Iris	150	4	Classification	[29]
Adult data set	1,605	123	Regression	[30]
CIFAR-10/ CIFAR-100	50000	$32 \times 32 \times 3$	Classification	[31]
Microbial hydrogen yield	35	5	Regression	**Publication I**
Acute Myeloid Leukemia	179	78	Classification	**Publication III**
Magnetoencephalography (MEG) data	756	408	Classification	**Publication V**

*There exists also 216x4000 data set named as ovarian cancer data.

2.3 Over-determined scenario

In contrast, an *over-determined* situation occurs when the number of observations is larger than the number of features, i.e., $n > p$. There are too many additional constraints to infer a solution. In general, over-determined systems have no solution if some of the equations are not a linear combination of each other or the equations are inconsistent [32]. The classical approaches, such as linear least squares [33] are used to find an approximate solution for over-determined problems [34]. Figure 2.4 (b) illustrates an over-determined system. If the solution is either non-unique or does not exist, then over-determined problems can be solved numerically by regularization approaches [25].

Consider the example mentioned in Section 2.2. Now, we ask 5 more persons, who also purchased apples and oranges from that grocery shop, the total price and the

number of the fruits they purchased. This information is more than sufficient to correctly interpret the unit prices of the fruits. In fact, asking only 2 persons is sufficient to obtain the result. This situation is an over-determined case and too many observations may reach the approximate solution if the data is noisy. Table 2.1 lists a few data sets of under-determined and over-determined scenarios.

2.4 Extremely under-determined scenario

In many active research areas including biomedical science and computational biology, the number of features, p is much larger than the number of observations, n. In a high throughput measurement technology, such as a microarray, a single measurement yields thousands to tens of thousands of gene expression levels. Nevertheless, the number of observations are few (say, tens) due to the high cost of the measurements [11]. High-dimensional data also arises in flow cytometry analysis for cancer diagnosis. In a single session, a flow cytometer can process from ten thousand to tens of thousands of cells and generate large quantities of measurement data. However, the observations are also limited to only a few patients [12]. These problems are termed as $p \gg n$ problems.

An approximate solution to a system of linear equations with n observations and p unknowns depends on the dimension p. The learning ability of a model can be improved by increasing the complexity of that model, and this complexity depends on the dimension p [35]. However, increasing $p \to \infty$ also requires n to be ∞ as well. Otherwise, the system becomes under-determined. This approach is known as *thermodynamic limit approach* [36, 37]. A wide variety of literature also exists that has studied the properties of p while p is growing or $p \to \infty$ [11, 38–41]. However, $p \gg n$ problems pose several challenges for many classical machine-learning approaches. The reason is that insufficient observations are available to estimate the underlying covariance structure properly [13]. Moreover, overfitting and the curse of dimensionality (see Section 2.5 and Section 2.6) are also a major concern. Dimensionality reduction and feature extraction approaches are already playing pivotal roles in this context [42]. Regularized approaches have also been used to confront the challenges in a $p \gg n$ setting [13].

2.5 Overfitting

One of the challenges in a machine-learning paradigm is the situation where a model starts to learn noise from the data. This phenomenon is considered as *overfitting*, since although the model is well approximated to a given limited set of data points, it may have a poor fit to the unobserved data set. Another limitation of the overfitted model is that it cannot be generalized to new observations. In contrast, *underfitting* occurs when the functions cannot capture the structure of the data adequately.

Figure 2.5 demonstrates an example of using linear regression to fit a curve. The plot shows a cosine function (blue curve) that we want to approximate with polynomial features. In addition, the observations from the true function (shown in circles) and the predicted curve from different models are shown. The models have polynomial features of three different degrees. We can see that a linear model with a polynomial of degree 1 is insufficient to fit the observations (green straight curve). This is referred to as *underfitting*. On the other hand, a polynomial of degree 20 gives an excellent fit to the data as the curve passes exactly through each data point (black curve). However, this model also gives a poor representation of the true function. This phenomenon is known as *overfitting*.

In this example, the polynomial of degree 4 approximates the true function perfectly and gives the best fit (red curve) among the models of three different degrees.

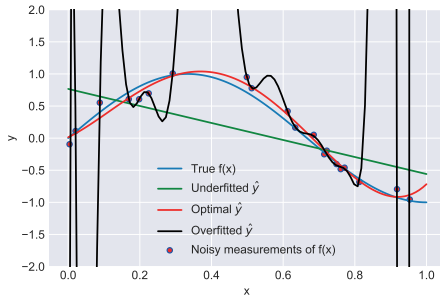


Figure 2.5: An example of fitting a curve. The measurements are generated from a cosine function (blue curve) with arbitrary noise. Three models with different degrees of polynomials are used to fit the curve using these measurements.

The danger of overfitting can be lessened by several approaches, such as regularization, model comparison, cross validation, dropout, early stopping, Bayesian priors and pruning. The fundamental idea of these approaches is to either explicitly penalize over-complex models or evaluate the performance of the model on test data that are independent of the training data.

The concepts of overfitting and underfitting are closely related to *generalization*. Generalization is the ability of a model to learn effectively from the training data and behave similarly for unseen data. In the underfitting zone, both generalization and training errors are higher (See Figure 2.6). As the model complexity increases (we can increase the model complexity, for example, by adding features and associated hyperparameters to the model), the number of training errors as well as generalization errors decreases. However, too complex models, as illustrated in Figure 2.5, may start to learn noise from the training data. At this point, we enter the overfitting region and the training errors continue to decrease. In contrast, the number of generalization errors starts to increase. Moreover, complex models have lower bias and higher variance. Underfitted models, in general, have a high bias and a low variance, whereas overfitted models have a low bias and a high variance. In machine learning, we want a model that both minimizes the errors for the observed data and generalizes well for unobserved data. Achieving these two objectives simultaneously is, however, difficult; a conflict known as the *bias – variance dilemma* [43].

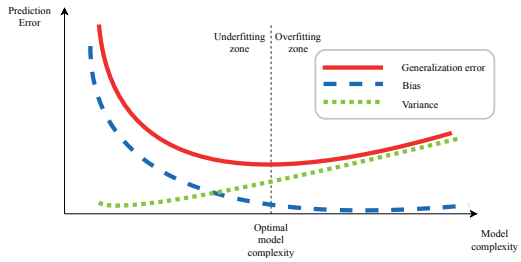
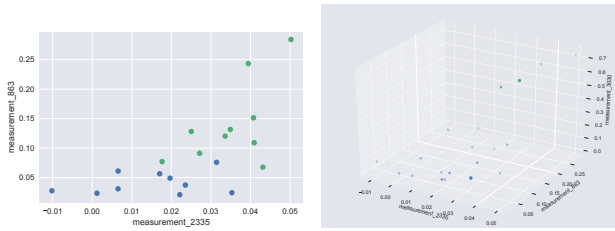


Figure 2.6: Model complexity vs error.

2.6 Challenges with high-dimensional data

Let us consider the first data set from Table 2.1. This data set contains information about 253 cancer patients, and controlled or healthy ones. We need to discriminate between the cancer patients and the healthy patients based on the 15,154 measurements for each patient. Thus, we can construct a design matrix of 253×15154. Here, $p \gg n$ and we need to solve the problem in a 15,154-dimensional space. To understand the curse of dimensionality, we can reduce the number of observations to 10 (5 samples from each class) and take two measurements, which we assume to be sufficient to discriminate between two classes. Figure 2.7 (a) shows that we cannot obtain a perfect separation between the cancer and healthy patients with two features. Adding another feature, for example, in Figure 2.7 (b), is sufficient to distinguish between the two classes.



(a) **(b)**

Figure 2.7: Classification with (a) two features, (b) three features.

This example suggests that increasing the number of features would result in perfect classification. However, adding a feature also increases the dimensionality of the feature space and the density of the observations becomes sparser. Due to this sparsity, one can easily find a separable hyperplane, as the probability of the training data lying on the wrong side of the best hyperplane is relatively low. Thus, it is hard to ascertain any meaningful conclusion without increasing the number of observations. The curse of dimensionality was first introduced by Bellman [44]. The performance of a model increases with the number of its dimensions until the optimal number of features is reached. Further increasing the dimensionality without increasing the number of samples results in a decrease in the model's performance (see Figure 2.8). Moreover, many machine-learning methods, such as the least squares method, require the number of observations to be larger than the number of features, i.e., $n > p$ explicitly.

There are several approaches for avoiding the curse of dimensionality. The aim is to reduce the dimension of the data, resulting in a better-fitted model. One way is to use feature-selection methods (such as filter, wrapper, and embedded methods [45]) to reduce the number of features and thus keep the dimension lower. Like feature selection, feature extraction or dimension reduction approaches can also be used. These methods are discussed in Section 3.4. Another way is to use regularized methods that are robust to high-dimensional settings, such as a support vector machine, lasso, and ridge regression [14–16].

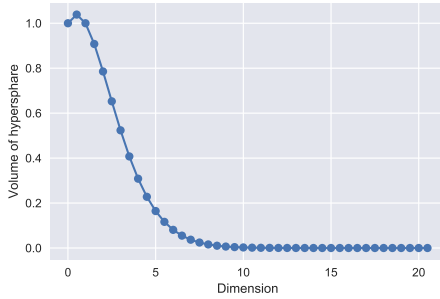


Figure 2.8: An illustration of the curse of dimensionality: the volume of hypersphere reduces to zero as the dimensionality increases.

3 Machine-learning principles

Machine learning, an emerging branch of artificial intelligence, explores the study and development of algorithms that can learn from *data* and make predictions on *data*. The aim is to make learning automated, accurate, reliable, and computationally fast. The algorithms utilize the observed data, also called *training data*, to discover patterns automatically and employ these learned patterns to make intelligent decisions on new or unseen data. The learning task can be generally divided into supervised and unsupervised learning. In supervised learning, patterns are learned by mapping from an input to a desired output, whereas in unsupervised learning, the pattern is learned directly from the input. In this chapter, we provide a brief overview of supervised and unsupervised learning methods. Next, we discuss extensions of the various approaches to regularization. Then, feature extraction, selection, and ranking methods are discussed as they play a vital role in understanding the underlying biological problems in question. We limit the discussion to the concepts relevant to the research work.

3.1 Supervised learning

A supervised learning problem can be thought of as a function approximation problem, given a vector $\mathbf{x} \in \mathbb{R}^p$ as input with p-dimensional feature space and $y \in \mathbb{R}$ as output or response variable. If there exists a functional relationship, defined by f, between the input and the output, then we try to approximate a function $\hat{f}$ such that $y = f(\mathbf{x})$. This approximation is generally learned from the training data $\mathfrak{D}_{\texttt{train}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where n is the number of observations. Once the learning is done, we can apply the trained model $\hat{f}$ to predict the response $\hat{y}$ for new or unseen $\mathbf{x}$ as $\hat{y} = \hat{f}(\mathbf{x})$. If y is quantitative (such as the price of a house or the height of a person), then the problem is known as a *regression* problem. On the other hand, if y is a qualitative variable, i.e., $y \in \{1, 2, \ldots, c\}$, where c represents a discrete number of classes, then the problem is known as a *classification* problem and the learned model is a *classifier*. Instead of discrete classification, we can also express our uncertainty of class membership for an observation $\mathbf{x}$ using a probabilistic distribution. This is known as *probabilistic classification*. In this classification problem, we model the likelihood functions or, more specifically, the class conditional densities given by $\Pr(\mathbf{x}|y = k)$, together with prior probabilities $\Pr(y = k)$ for the classes. Then we compute the posterior probabilities using Bayes' theorem,

$$\Pr(y = k|\mathbf{x}) = \frac{\Pr(\mathbf{x}|y = k)\Pr(y = k)}{\Pr(\mathbf{x})}. \tag{3.1}$$

Here, $\Pr(\mathbf{x}) = \sum_{i=1}^{c} \Pr(\mathbf{x}|y = i)$.

In order to choose the best approximation in supervised learning, one can measure a *loss function* $L(y, \hat{y})$ that measures the deviation between the true response y and the predicted

response $\hat{y}$ from the learned model. Examples of commonly used loss functions are listed in Table 3.1. In regression, squared loss and absolute loss functions are commonly used. On the other hand, in classification, any of the loss functions listed in Table 3.1 can be used if we assume one-hot encoding i.e., a binary classification with the response $y \in \{0, 1\}$. Note that, in this case, the squared loss and the absolute loss reduce to the zero-one loss case.

Table 3.1: Examples of loss functions L. Adapted from [46].

Loss function	Definition
Zero-one loss	$\begin{cases} 0, & \text{if } y = \hat{y} \\ 1, & \text{otherwise.} \end{cases}$
Squared loss	$(y - \hat{y})^2$
Absolute or Laplace loss	$\lvert y - \hat{y} \rvert$
Logistic loss	$\frac{1}{\log(2)} \log(1 + \exp^{-y\hat{y}})$

Figure 3.1 illustrates some loss functions in classification problem, where the true responses for the data are $\mathbf{y} = (1, 1, \ldots, 1)$. In this example, we assume that the values of predicted response $\hat{y}$ by a classifier are $-3 \leq \hat{y} \leq 5$. Note that the loss functions are convex upper bound on the zero-one loss and they all have a loss penalty of 1 at $\hat{y} = 0$ [13, 16, 46, 47]. In this example, we implicitly assume that the prediction of the model is either a probability (between 0...1) or a class indicator {0, 1}.

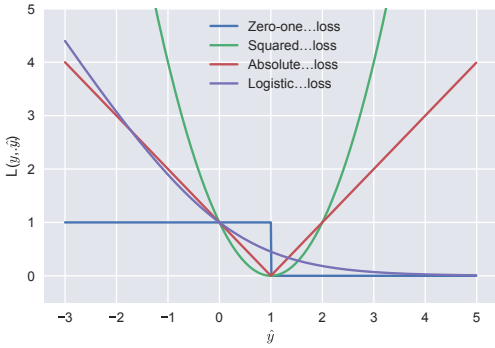


Figure 3.1: Loss functions in classification problems when the true class label, $y = 1$.

3.1.1 Linear models for regression

The functional relationship between input and output for the model $y = f(\mathbf{x})$ can be either linear or nonlinear. Linear models play a special role in machine learning as they are simple and the relationship between the response y and the input features $\mathbf{x}$ is assumed to be linear. Hence, they are easy to interpret and their statistical properties are well-studied [48, 49]. The linear model can be fitted using the *linear regression* approach such that the response is a linear combination of feature variables, i.e.,

$$y = \beta_0 + \mathbf{x}^T \boldsymbol{\beta} + \epsilon. \tag{3.2}$$

Here, $\mathbf{x}^T \boldsymbol{\beta}$ is the inner product between $\mathbf{x}$ and $\boldsymbol{\beta}$. Eq. 3.2 represents a linear model, where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$ is a parameter vector in p-dimensional space, β_0 is a constant bias,

and ϵ is a random error independent of $\mathbf{x}$. The ϵ represents the errors in the relationship and we assume ϵ follows Gaussian distribution with zero mean and standard deviation. In general, we have a set of training data $\mathfrak{D}_{\texttt{train}}$ from which we can learn the vector of model parameters or coefficients, $\boldsymbol{\theta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$. Here, $\boldsymbol{\theta}$ is a $(p+1)$-dimensional parameter vector that includes β_0 and $\boldsymbol{\beta}$. On the other hand, nonlinear models are an extension of linear models, where we assume f to be nonlinear. Let us construct a design matrix, $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$, and an output vector $\mathbf{y}$. Each row of $\mathbf{X}$ represents an augmented vector generated from input features $\mathbf{x}_i$ such that $\mathbf{X}_i = (1, \mathbf{x}_i^T)^T$. The y_i is the ith element of $\mathbf{y}$ and corresponds to $\mathbf{x}_i$ from the $\mathfrak{D}_{\texttt{train}}$ set. We can then rewrite Eq. 3.2 in a vector form as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta}. \tag{3.3}$$

With only one feature variable, the process is known as *simple linear regression* and with more than one feature, the process is known as *multiple linear regression* [50]. The most popular approach to estimate $\boldsymbol{\theta}$ is referred to as the *ordinary least squares* method, where we pick the parameters to minimize the residual sum of squares (RSS) or the loss function for Eq. 3.3 such that

$$RSS(\hat{\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2. \tag{3.4}$$

Here, $\|.\|$ is the standard ℓ_2-norm in the p-dimensional Euclidean space $\mathbb{R}^p$. We can express ℓ_2-norm of $\mathbf{x}$ as $\|\mathbf{x}\|_{\ell_2} = (\,|x_1\,|^2 + \,|x_2\,|^2 + \ldots + \,|x_p\,|^2)^{1/2}$. Eq. 3.4 has a unique solution provided that the columns of $\mathbf{X}$ are linearly independent and hence $\mathbf{X}^T\mathbf{X}$ is positive semidefinite. This solution can be expressed as

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.5}$$

The least squares estimator employs the properties of the *best linear unbiased estimator* (BLUE), where the estimate of $\boldsymbol{\theta}$ has the smallest variance of all linear estimators with no bias. This is asserted by the *Gauss-Markov theorem* [51]. However, least squares are sensitive to outliers. Moreover, if the design matrix $\mathbf{X}$ is non-full rank, for example, if two or more features are linearly dependent, then $\mathbf{X}^T\mathbf{X}$ is singular and hence $\hat{\boldsymbol{\theta}}$ is not uniquely defined. Deficiency in rank can also occur with a small number of training data in high-dimensional settings, i.e., $p \gg n$. There are several approaches, such as shrinkage and subset selection strategies to control the variance, and the approaches relevant to this thesis will be discussed in the following section. An improvement can be achieved in the overall prediction accuracy of the learned model by allowing a little bias and reducing the variance. Hence, the model can be easily interpreted and has the potential for lower prediction error than the model that considers all possible combinations of features.

3.1.2 Linear models for classification

In classification problems, the task is to take an input or feature vector $\mathbf{x} \in \mathbb{R}^p$ and assign it to one of c discrete classes. It is also possible to assign an input to multiple classes (known as multilabel); nevertheless, here, we focus on mutually exclusive classes (multiclass). In a multiclass problem, the classes are disjointed so that each input is assigned to one and only one class. A classifier achieves this by dividing the feature space, denoted by $\mathbb{F}$, into *decision regions* $\mathcal{R}_i, i = 1, 2, \ldots, c$ such that $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$, if $i \neq$

j and $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \ldots \cup \mathcal{R}_c = \mathbb{F}$. The feature space, $\mathbb{F}$, is a p-dimensional Euclidean space $\mathbb{R}^p$, which includes all the features. The linear classifier defines a *discriminant function* $g(\mathbf{x})$, which assigns $\mathbf{x}$ to class c_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$, $i \neq j$. The decision regions are separated by boundaries, known as *decision boundaries* or *decision surfaces*. Of course, $g_i(\mathbf{x}) = 0$ in the decision boundaries. In linear models for classification, we consider that the decision surfaces are linear and therefore, they can be defined by $(p-1)$ dimensional hyperplanes with a p-dimensional input space. In this case, the classes are said to be *linearly separable*. In the following, we will introduce two such linear models for classification.

3.1.2.1 Linear discrimination analysis

Linear discriminant analysis (LDA) is a classical method in the classification problem domain originally introduced by Fisher in 1936 [29]. The idea is to find, in a binary classification problem, for example a projection line that separates the two classes as much as possible. In other words, a projection vector $\hat{\boldsymbol{\beta}}$ is selected that maximizes the ratio of the variances of *between-class* to the *within-class* such that

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \frac{(\boldsymbol{\beta}^T(\mathbf{m}_1 - \mathbf{m}_2))^2}{\boldsymbol{\beta}^T \mathbf{S}_W \boldsymbol{\beta}}. \tag{3.6}$$

Here, we assume $g(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta} + \beta_0$ for a binary classification problem and $\mathbf{m}_1 \in \mathbb{R}^p$ and $\mathbf{m}_2 \in \mathbb{R}^p$ are the class means and $\mathbf{S}_W \in \mathbb{R}^{p \times p}$ is the pooled within-class sample covariance matrix. This ratio describes how well the classes are separated. The maximum separation occurs when $\boldsymbol{\beta} \propto \mathbf{S}_W(\mathbf{m}_1 - \mathbf{m}_2)$. In order to use Fisher's discriminant as a linear classifier, we need to determine a discriminant function $g(\mathbf{x})$ and specify a threshold, β_0, so that we can assign $\mathbf{x}$ to class c_1 if $g(\mathbf{x}) > \beta_0$ and to class c_2, otherwise. There is no general rule for choosing the threshold. One approach can be taking the average of the projected class means. Alternatively, a decision theoretic approach, *Neyman-Pearson lemma*, can be used [52]. In general, the data to be discriminated are projected on to $\boldsymbol{\beta}$ and the threshold that best separates the data is chosen. If the data are normally distributed with a homoscedasticity assumption, then we can compute the threshold β_0 explicitly [47] such that

$$\beta_0 = -\frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)^T \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) - \log\left(\frac{\mathbb{P}\mathrm{r}(c_2)}{\mathbb{P}\mathrm{r}(c_1)}\right). \tag{3.7}$$

Here, $\mathbb{P}\mathrm{r}(c_i)$ is the prior probability of class c_i. Fisher's ratio can also be generalized to multiclass problems [47].

3.1.2.2 Logistic regression

Similar to the LDA, logistic regression estimates the linear decision boundaries, but differs in the learning algorithm. Logistic regression is *probabilistic* in nature, which differentiates it from other linear models. Instead of predicting straightforward class memberships, logistic regression models the class probabilities. Cox was one of the pioneers in developing this method [53, 54]. However, the foundation of logistic regression, which relies on a *logistic function*, dates back to the work of Pierre-François Verhulst in the early 19^{th} century. Verhulst used the logistic function to model the growth of human population [55]. The function is a sigmoid function, characterized by an "S" shaped curve, that can be expressed as

$$\eta(t) = \frac{\exp(t)}{1 + \exp(t)} = \frac{1}{1 + \exp(-t)}. \tag{3.8}$$

Here, $\eta(t) \in (0, 1)$, for all $t \in \mathbb{R}$. A graph of the logistic function for $-5 \leq t \leq 5$ is illustrated in Figure 3.2.

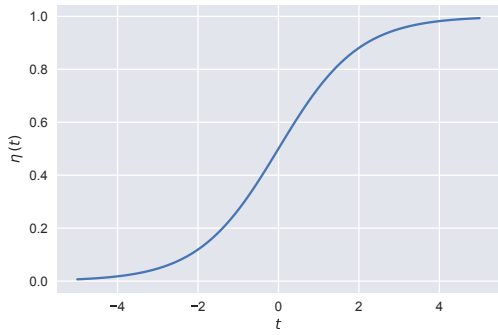


Figure 3.2: The logistic sigmoid curve.

A logistic regression model uses the logistic function to map the projection of $\mathbf{x}$ to an estimate of the probability that $\mathbf{x}$ belongs to a class. In a two-class problem, the logistic regression model can be defined for a feature vector $\mathbf{x} \in \mathbb{R}^p$ belonging to class 1 with the probability

$$\Pr(c = 1 \mid \mathbf{x}) = \frac{\exp(\beta_0 + \mathbf{x}^T\boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}^T\boldsymbol{\beta})} = \frac{1}{1 + \exp(-(\beta_0 + \mathbf{x}^T\boldsymbol{\beta}))} \tag{3.9}$$

and to class 2 with the probability

$$\Pr(c = 2 \mid \mathbf{x}) = 1 - \Pr(c = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(\beta_0 + \mathbf{x}^T\boldsymbol{\beta})}. \tag{3.10}$$

A logistic regression model is trained using *maximum likelihood estimation* (MLE) to learn the regression coefficients. Since finding a closed-form solution for maximizing the likelihood function is not possible, numerical search methods (such as the Newton-Raphson method) are used to compute the MLE. Alternatively, *iteratively reweighted least squares* (IRLS) can be used to solve MLE problems [56].

This method models the odds of an outcome and, therefore, it has become a viable tool in statistical analysis, particularly in applications, where binary responses occur frequently. For example, patients survive or die, have cancer or not. Depending on the type of the class labels, logistic regression can be categorized into three types: binomial, multinomial, and ordinal. In binomial, the observations can have either of two possible class labels: "0" and "1", which can represent, for example, "yes" vs. "no", "disease" vs. "controlled". The multinomial logistic regression is used when the data have unordered three or more possible class labels, such as mode of transport with "bus" vs. "car" vs. "train" vs. "walking". The ordinal logistic regression deals with ordered class labels, such as test scores with "excellent" vs. "average" vs. "poor". Logistic regression can be thought of as a special case of a traditional neural network with a single neuron [57].

3.1.3 Tree-based models

Trees, particularly binary trees, are an attractive and powerful data structure in the computer science field due to their computationally efficient construction and utilization.

A decision tree is a tree-like structure with a collection of decision nodes and their possible consequences. A decision tree offers an interesting and often illuminating way of exploring the data in a supervised learning problem. The tree learns a classification or a regression model from $\mathfrak{D}_{\texttt{train}}$ by dividing the feature space recursively into rectangular regions and assigns a class label for each region. The data is broken down into smaller and smaller subsets, and simultaneously an associated tree structure is developed. The final outcome is a tree with a collection of decision nodes and leaf nodes. The decision nodes in the tree correspond to input features and the leaf nodes represent class labels (in classification) or continuous values (in regression). Each interior node in the decision tree compares feature values with a selected threshold and forms a hierarchical structure of if-then rules. Figure 3.3 illustrates the concept of a decision tree using a simple example. A decision "Should I take an umbrella?" and three features {sky condition, humidity, windy}. The decision tree will select the best feature, for example, the 'sky condition' and create a branch for possible values, such as 'sunny' and 'cloudy' in this case. The observations are then passed to the appropriate branch based on their value for the feature being tested. If the observations at a child node belong to the same class, that node becomes a leaf node. Otherwise, the procedure is repeated recursively for each new child node. More in-depth discussion of the topic can be found in [58].

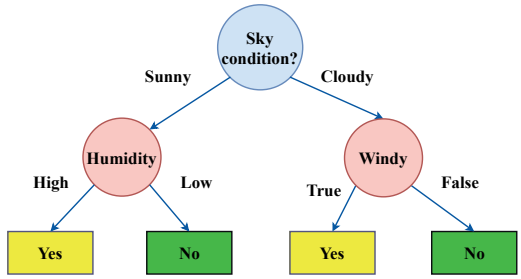


Figure 3.3: Example of a decision tree.

A popular decision tree algorithm, *classification and regression trees* (CART), has been proposed by Breiman et al. [59]. The CART provides a basis for decision tree algorithms to solve both classification and regression predictive modeling problems. One of the advantages of using CART is that it can handle both numerical and categorical data. Moreover, the method is nonparametric and nonlinear in nature [60]. However, CART algorithms can build overly complex trees which are prone to overfitting. Moreover, a small variation in the training data may generate a completely different series of splits and the effect of an error in an internal node can propagate all the way down to the child nodes [59]. This makes the decision tree unstable. In addition to the lack of generalization and stability, in order to find the best feature during each split operation, a greedy selection is performed using impurity measures, such as Gini index, misclassification error, cross entropy or deviance and least squares. The greedy selection is not guaranteed to achieve globally optimal decision trees.

Ensemble learning methods, such as boosting [61] and bagging [62], are introduced to deal with the aforementioned challenges. These methods generate many weak classifiers and then aggregate their results. In bagging, for instance, each tree is constructed independently using a random bootstrap subsample of the training data and then the results are combined using a simple vote. In boosting, on the other hand, additional weights are applied to misclassified errors during the construction of successive trees. The

results are then combined using a weighted vote.

A parallel ensemble method, known as *Random forest*, adds additional randomness to the bagging method. The method was originally proposed by Breiman [63]. A random forest is a collection of several decision trees built by bootstrap aggregation. M trees are constructed from M subsets $\mathfrak{D}_{\texttt{train}}{}' \subseteq \mathfrak{D}_{\texttt{train}}$ of the training samples. These bootstrap samples are drawn uniformly at random, with replacement, from $\mathfrak{D}_{\texttt{train}}$ to learn different trees. The random forest is attractive as it is tolerant to overfitting. Moreover, it has the ability to efficiently rank features [64]. Both the interpretation of the model and the construction of a good parsimonious model require feature selection, where the aim is to identify relevant features for prediction. Feature selection algorithms often use feature ranking as a selection strategy. Feature ranking is a statistical measure that assigns a score to each feature. Random forest includes two strategies for feature selection: Gini importance and permutation importance. The former strategy decreases the weighted impurity of a tree during training whenever a feature is selected for splitting a decision node. For classification, the weighted impurity is either a Gini impurity (an impurity measure based on the Gini index) or an information gain. For regression, this impurity is the minimization of the sum of squares. The latter measures the effect of model accuracy on the permutation of values for each feature. The feature importance measures, however, are biased, particularly in a $p \gg n$ problem [65]. Additionally, permutation importance overestimates the feature importance with highly-correlated features [66]. Furthermore, the feature importance scores do not indicate direct separation between the relevant and irrelevant features [67].

For the above-mentioned reasons, an improvement in the random forest framework is proposed by Tuv et al. [67]. The *Random Forests with Artificial Contrast Ensembles* (RF-ACE) framework expands the original data with artificially-generated contrast variables, which are independent from the response variable. The ranks of these variables are used to remove the irrelevant features. The idea is that any stable feature selection method will produce a higher score for true relevant features over an artificially-generated feature, since the artificial feature is irrelevant to the response variable. The prediction accuracy of RF-ACE has also been improved by using gradient-boosting tree methods [67].

3.2 Unsupervised learning and clustering

This section concentrates on unsupervised learning, where we have no mapping from input to output, i.e., the data are unlabeled. This is analogous to providing a one-year old child having no prior knowledge of colors with a set of colored objects and requesting the child to sort those objects according to color. The aim of such unsupervised learning problems may be to seek similarity within the objects. There are several reasons for one to be interested in unsupervised learning. CIFAR-10/100, for example, mentioned in Table 2.1 is a collection of labeled subsets from 80 million tiny images [68]. This overwhelming collection of images has few or no class labels at all. In this case, manual label annotation is an impractical solution. Therefore, a classifier can be constructed on a small labeled data set and then the label information can be propagated to the unannotated large data set in an unsupervised manner. Moreover, we can gain insight into the structure of the data by discovering similar patterns. In addition, unsupervised learning can be used as a preprocessing step, for instance, to learn features within the same class. In order to learn features, we can either increase dimensions by sparse representation [69] or reduce dimensions by lower-dimensional representation [70], before training.

Clustering is the process of grouping data into sets of disjoint classes called *clusters* [71]. Data expressed in a similar way are grouped within the same class. Clustering methods can be hierarchical and nonhierarchical. In hierarchical clustering, data are grouped into clusters by specifying the relationships among the data in a cluster. In contrast, in a nonhierarchical approach, clusters are formed without specifying the relationships between the data.

The nonhierarchical approach is based on iterative relocation, which involves a number of learning steps to find an optimal solution for dividing gene expression data [72]. Such an approach requires a prior knowledge of the number of clusters. The clustering is then performed by grouping the existing objects into these predefined clusters. The *K-means* algorithm and *Kohonen Self-organizing Map* (SOM) are examples of a nonhierarchical approach [73]. In K-means clustering, the average expression profile is built for each cluster in the initial step. The objects are reattributed based on the proximity of an expression profile, and a new cluster is built. The procedure is performed for a fixed number of iterations until the clusters converge to a stable state [73]. Of course, the number of clusters must be specified initially. One option is to choose this number randomly. Another way is to estimate it by first performing a hierarchical clustering of the gene expression data. The SOM method is fast and simple to implement. Unlike K-means, the orientation of the clusters needs to be specified in SOM. The method assumes that each object is a node of a two dimensional grid space. The position of the nodes is redefined at each successive step and reformed to fit the data [73].

In contrast, the hierarchical approach is the best-known method for analyzing gene expression microarray data. This clustering approach generates a hierarchical series of nested clusters, which can be graphically viewed by a tree, known as a *dendrogram*. Each branch of the dendrogram represents the formation of clusters as well as the significance of any similarity between the clusters. A predefined number of clusters can be achieved by cutting the dendrogram at a certain level [74]. The hierarchical approach can be further classified into agglomerative and divisive methods. The agglomerative method is a top-down approach, whereas the divisive method is a bottom-up approach. In the case of agglomerative methods, each object is assumed to be a cluster and iteratively merges the closest pair of clusters until all objects are in one cluster. In contrast, divisive methods start with the assumption that all objects belong to one class and iteratively split a cluster until all objects are separated into a single cluster [73, 74].

In agglomerative approaches, the similarities between clusters are measured using several strategies, such as single linkage, complete linkage, and average linkage. These strategies can be differentiated by measuring the distance between two clusters. In single linkage clustering, the distance between clusters is the minimum. Conversely, in complete linkage clustering, the distance between clusters is the maximum. The single linkage method is insensitive to outliers, whereas complete linkage is sensitive to outliers. The average distance between all possible cluster pairs is taken into account in the average linkage method [73]. There is no theoretical guideline for choosing the best linkage method. However, researchers usually prefer the average linkage method. The divisive method is the opposite of the agglomerative approach. The splitting of objects into clusters at each step is decided either by principle component analysis or by graph theoretical methods [74].

Hierarchical clustering methods have been widely used in many research areas including the classification and modeling of gene expression data. For instance, several studies have performed molecular classification on cancer cells. Eisen et al. have used this method

to group genes with a similar expression function in *Saccharomyces cerevisiae* [71]. In addition, Alon et al. have used the divisive method for gene expression analysis [75]. Even so, the algorithm fails with a large number of genes as the data increases in complexity. Although it is not sensitive to outliers, the method is lacking in robustness. As a result, this method is not feasible for the analysis of large data sets.

The K-means algorithm has several drawbacks. Although the method requires the number of gene clusters to be predefined in a given expression data set, it is unknown in advance. Consequently, the algorithm needs to be run repeatedly to obtain an optimal number of gene clusters. However, this may not be a practical solution for large gene-expression data. Besides, large data contain a significant amount of noise, which severely affects the accuracy of the algorithm [74]. Conversely, SOM is robust to noisy gene expression data. The method produces reliable partitions in the presence of noise in data. However, Thalamuthu et al. showed that SOM has worse performance in identifying interesting patterns due to the merging into one cluster. Therefore, the method needs to be used with caution [75].

The selection of an appropriate method is a challenging task in analyzing gene expression data. The results may vary depending on methods with different parameters. Thalamuthu et al. have demonstrated the feasibility of the methods for the clustering of gene-expression data [75]. A good understanding of the problem domain, such as the availability of clustering options influences and/or facilitates the selection of the best method. Additionally, the knowledge of biological aspects helps in choosing a tool that satisfies certain requirements, for example, the capacity to exclude the outliers. Furthermore, several studies suggest different criteria of acceptability for the algorithms. For instance, hierarchical approaches are insensitive to duplicate data samples, whereas nonhierarchical approaches are feasible for large amounts of data [72]. Clustering validation can be another way for selecting an appropriate method. The quality of clusters can be verified by the *Dunn's based indices* approach. In addition to measuring the quality, the validity can be measured by the *silhouette* [72] method. We exploit some of these clustering properties in **Publication II**.

3.3 Regularization

The function approximation is often an *ill-posed* problem. Ill-posed problems do not satisfy three properties of a *well-posed* problem, formulated by Jacques Hadamard, a French mathematician, in 1902 [76]. According to Hadamard, the ill-posed problems may not have solutions; no unique solution exists or the solutions are unstable. One way to overcome the ill-posed nature of a problem is to introduce additional information, the process is known as *regularization*. Regularization is a strategy that adds additional constraints to a machine-learning model. These additional constraints can be, for example, restricting the parameter values, or combining several hypotheses to explain the data in order to overcome the issue of overfitting and to promote generalization [77]. A simple form of regularization assumes that the features are independent within each of the classes, so the within-class covariance matrix is diagonal [13]. A regularization term or regularizer is a function, $\Omega(\boldsymbol{\beta})$ that can be added to the loss function such that

$$J(\boldsymbol{\beta}) = L(y, f(\mathbf{x}, \boldsymbol{\beta})) + \lambda\Omega(\boldsymbol{\beta}), \tag{3.11}$$

where $\lambda \in [0, \infty)$ is a hyperparameter that controls the strength of the regularizer. If $\lambda = 0$, then there is no regularization. Larger values of λ lead to more regularization. In general, $\Omega(\boldsymbol{\beta})$ is chosen to impose a penalty on the complexity of the model, where

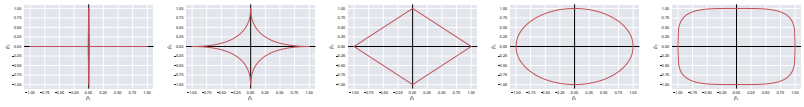
constraints include restrictions for smoothness and boundaries on the vector space norm. Regularization was first proposed by Tikhonov and Arsenin [78] for the inverse problem. The goal was to add a regularization factor $\lambda \mathbf{I}$ to stabilize an ill-posed under-determined scenario.

The standard regularization methods are used to learn a simpler model by adding a penalty for the size of the model parameters. The norm penalty includes ℓ_1 and ℓ_2 regularizations. In ℓ_1 regularization, the penalty, $\Omega(\boldsymbol{\beta}) = \sum_{i=1}^{p} |\beta_i|$ is added to regularize the loss function. The ℓ_1 penalty causes a subset of the parameters to become zero. This encourages the discarding of the corresponding parameters and makes it a natural candidate for feature selection by generating a sparse vector of parameters. Section 3.3.2 discusses a well known example of ℓ_1 regularization, lasso, which integrates the penalty with the linear least squares. Another variation of ℓ_1 is used with logistic regression [79]. The ℓ_1 regularization has also been extended to other approaches, such as multi-layer perceptrons, and support vector machines [80].

On the other hand, in ℓ_2 regularization, a $\Omega(\boldsymbol{\beta}) = \sum_{i=1}^{p} \beta_i^2$ penalty is added to the loss function. More discussion on this topic can be found in Section 3.3.1. An adjustment between ℓ_1 and ℓ_2 regularizations, known as *Elastic net*, is proposed by Zou and Hastie [81]. In this case, the penalization term encourages the sum of both the absolute values and the square of the parameters to be small such that

$$\Omega(\boldsymbol{\beta}) = \gamma \|\boldsymbol{\beta}\|_{\ell_1} + (1-\gamma) \|\boldsymbol{\beta}\|_{\ell_2}. \tag{3.12}$$

Here, $\gamma \in (0, 1)$ is a mixing hyperparameter to determine the proportions between ℓ_1 and ℓ_2 regularizations. In addition to this, a generalization of penalized regularization, ℓ_q, has been introduced by Frank and Friedman [82], where $\Omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_{\ell_q}$. This is also known as *bridge* regression. Figure 3.4 illustrates the contour regions for ℓ_q regularization with different values for q in two dimensions. The value $q = 1$ corresponds to lasso. Ridge regression corresponds to $q = 2$. For $q > 1$, the regularization encourages the selection of features at the group level instead of the individual level.



(a) $q = 0.1$ **(b)** $q = 0.5$ **(c)** $q = 1$ **(d)** $q = 2$ **(e)** $q = 4$

Figure 3.4: Examples of isosurfaces of the regularization term for differnent q. Illustration adapted from [13].

The strategies of regularization have been extended in the machine-learning field to cope with challenges, such as overfitting and generalization. These strategies include data augmentations, dropout, adversarial training, etc. An excellent discussion on regularization can be found in [77].

3.3.1 Ridge regression

The least squares solutions can become unstable since the method is highly dependent on $\mathfrak{D}_{\texttt{train}}$. This instability can lead to overfitting, already discussed in Section 2.5, and hence a model with poor predictive performance may be selected from a set of competitive

models. Adopting a poor model may have invalid inference and mislead the underlying process that generates the data. Although a least square solution may result in unbiased mean squared error (MSE), it is uncertain that this MSE is the smallest one. There may exist a biased estimator with a smaller MSE. Such an estimator can add a bias for a large reduction in variance. The ill-posed nature and overfitting can be overcome by shrinking a subset of least squares coefficients. Such a method is *Ridge regression*. This method was first introduced in statistics with the aim of alleviating the problem of singular $\mathbf{X}^T\mathbf{X}$ [14]. A penalization term is added to the minimization problem of Eq. 3.4,

$$\hat{\boldsymbol{\theta}}_{ridge} = \arg\min_{\boldsymbol{\theta}} \{\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2\}. \tag{3.13}$$

Here, $\lambda \geq 0$ is a scalar hyperparameter that controls the amount of shrinkage. The hyperparameter λ shrinks the coefficients towards zero. For a larger value of λ, the amount of shrinkage is greater. In an inverse problem, the role of this penalization term is to alleviate the ill-posed problem and make the solution unique; this process being known as *Tikhonov regularization* [78]. In neural networks context, this approach is known as *weight decay*. The minimization problem of Eq. 3.13 has a closed form solution [13], which is expressed as

$$\hat{\boldsymbol{\theta}}_{ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{3.14}$$

where $\mathbf{I}$ is a $p \times p$ identity matrix. This solution adds a positive constant to the diagonal of $\mathbf{X}^T\mathbf{X}$ and this, in turn, alleviates the singularity problem during inversion process.

3.3.2 Lasso

Similar to ridge regression, *Least Absolute Shrinkage and Selection Operator (Lasso)* is an alternative approach for shrinking the regression coefficients. However, the lasso penalty uses ℓ_1-norm instead of the ℓ_2-norm used in the ridge penalty. We can write the lasso problem in *Lagrangian form* as

$$\hat{\boldsymbol{\theta}}_{lasso} = \arg\min_{\boldsymbol{\theta}} \{\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\beta}\|_{\ell_1}\}. \tag{3.15}$$

Lasso was proposed by Tibshirani [15] and in the signal processing literature this method is also known as *basis pursuit* [83]. Unlike ridge regression, there is no closed form expression for Eq. 3.15 as the term $\|\boldsymbol{\beta}\|_{\ell_1}$ makes the solution nonlinear in $\mathbf{y}$. However, there are many effective algorithms and quadratic programming available to solve the lasso problem [84, 85]. For instance, least angle squares is an iterative approach and moves towards the minimum step by step. This method computes the solutions for all values of $\lambda \in (0, \infty)$ [85]. Using the ℓ_1 penalty, lasso models yield sparse solutions by shrinking the model coefficients towards zero. Thus, lasso acts as a variable selection approach which, in turn, makes it an attractive alternative for the practitioners.

Figure 3.5 presents an example of a lasso penalty (green color) and a ridge penalty (yellow color) with respect to two model parameters β_1 and β_2 in a 2-dimensional setting. The elliptical contours represent the RSS. The least squared solution is at the minimum of the surface represented by the elliptical contours and marked with the red '$\star$' sign. Both lasso and ridge regression minimize Eq. 3.4, however, with respect to different constraints:

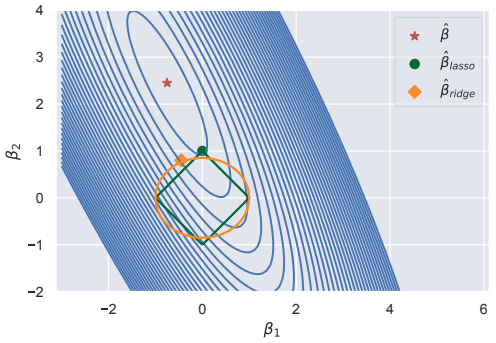


Figure 3.5: Visualization of the lasso and ridge penalties. The area inside the green isosurface is the constraint region $|\beta_1| + |\beta_2| \leq 1$ for lasso and the area inside the yellow isosurface is the constraint region $\beta_1^2 + \beta_2^2 \leq 1$ for ridge regression. The residual sums of squares are illustrated by the elliptical contours. The innermost elliptical contour has a smaller residual sum of squares [86].

$|\beta_1| + |\beta_2| \leq s$ for lasso and $\beta_1^2 + \beta_2^2 \leq s$ for ridge regression. Here, s is a constant value. For lasso, the constraint is a diamond shaped region while for ridge, it is a circle shaped region. A feasible set of solutions can be found within these regions whenever the elliptical contours touch them. For lasso, the constraint region is a square with corners on the coordinate axes, where all but one parameter are exactly zero. Therefore, the contours may touch the squared region either in a corner or on an edge between corners with some of the parameters being exactly zero. On the other hand, there are no corners in the constraint region for the ridge regression solution. Hence, a solution with parameters set to exactly zero rarely occurs [15, 86]. This sparseness makes the lasso an embedded feature selection tool. For this reason, in **Publication I**, **Publication III**, and **Publication IV**, we exploit this property of the method.

3.3.3 Regularization with generalized linear models

Logistic regression is an example of *generalized linear models* (GLMs) which represent a generalization of linear and nonlinear models. This framework was proposed by Nelder and Baker [87], where the response variable distribution must be a member of exponential families. In logistic regression, for instance, y is a member of the Bernoulli distribution. Other members of GLMs are normal, Poisson, binomial, exponential, and gamma distributions. We consider a generalization of Equation (3.2), where we transform the linear function of $\boldsymbol{\beta}$ using a nonlinear function $\phi(.)$ such that

$$\mathbb{P}\mathrm{r}(y = 1|\mathbf{x}) = \phi(\beta_0 + \mathbf{x}^T\boldsymbol{\beta}). \tag{3.16}$$

Here, $\phi(.)$ is known as an *activation function* and its inverse is known as a *link function*. Although the relationships between the response and the features are no longer linear due to the presence of a nonlinear function, the decision surfaces are still linear functions of $\mathbf{x}$. The estimation of the parameters is usually done by MLE. A regularized version for a broader class of GLM framework has been proposed by Friedman et al. [88], where the penalization term is added to the log-likelihood loss function. The authors implemented a `glmnet` algorithm that is computationally fast, particularly in a $p \gg n$ setting. The

algorithm includes linear regression, binomial and multi-nomial logistic regression methods along with ℓ_1, ℓ_2 and elastic net regularizations. Unless otherwise stated, the `glmnet` algorithm is used for parameter estimation in this thesis .

3.4 Feature extraction, selection, and ranking

The basic idea of feature extraction is to transform the raw data into a more manageable representation for the subsequent learning process, yet the representation is still able to accurately and completely describe the original data. The transformation process can be, for example, reduction of the dimension of the data [89]. Feature extraction plays a key role in model construction. Several dimension-reduction approaches exist in the literature. For instance, *principal component analysis* (PCA) is a well known linear transformation approach for dimensionality reduction [90]. For nonlinear transformation kernel-PCA can be used [91]. Another dimension reduction method is linear discriminant analysis (LDA), which uses the second-order statistical information, i.e., covariances. This, however, limits the number of features that can be generated [92]. Independent component analysis (ICA) is another interesting tool for dimension reduction that tries to find independent features by maximizing the statistical independence of the features [93, 94]. Other variations of feature extraction strategies are discussed in **Publication III** (see Table 2 of **Publication III**).

During model construction, one of the important tasks is the selection of a subset of relevant features or variables. This selection process is known as *feature selection*. The selection of relevant features has several advantages. First, the identification of relevant features simplifies the models and makes them easier to interpret. Second, removing the features that do not contribute to the model design also reduces the dimensionality of the feature space. Third, this selection process improves the training efficiency by reducing time and storage requirements. Finally, this promotes the generalization ability of models by reducing the risk of overfitting. The use of feature selection is motivated by various application domains, such as text mining, image processing, computer vision, biomarker discovery in bioinformatics and fault diagnosis in industrial applications [95].

Feature selection can be a part of the data preprocessing step as well as the model construction process. The methods can therefore be classified as filters, wrappers, embedded, and hybrid methods [45]. The selection of features in filter methods is based on the statistical properties of the features and the process is independent of modeling algorithms. Some of the common statistical measures are Pearson's Correlation, Chi-Square, and Fisher's score. The wrapper methods are dependent on modeling algorithms and are more computationally demanding than filter methods. Wrappers use a subset of features and evaluate them based on the model performance. The hybrid methods combine the properties of filter and wrapper methods. In embedded methods, feature selection becomes the part of the model construction.

This thesis considers the embedded methods for selecting relevant features as they include the feature selection as a part of the training process. This can be more efficient in several ways. The available data are utilized in an efficient way without having to split the training set into training and validation sets. Moreover, since the retraining of the model can be avoided for each feature subset in question, the solution can be computationally efficient in terms of time and memory requirements. In addition, many machine-learning algorithms have built-in mechanisms to perform feature selection [96]. The most popular examples of embedded methods are *Lasso* and *Elastic net*, which use regularization to

reduce overfitting [81, 97]. Lasso regularizes the weights of the features with the ℓ_1 penalty and shrinks those weights that do not contribute to the learning model to zero. Elastic net combines the ℓ_1 penalty of lasso and the ℓ_2 penalty of ridge regression. Another popular approach is the *Recursive Feature Elimination* (RFE) algorithm, commonly used with Support Vector Machines (SVM) [98]. In this case, a model is trained at each iteration with all current features, and ranking criteria are evaluated for each feature. The feature with the lowest ranking criterion is pruned from the current feature set. This process is repeated on the pruned set until the desired number of features is reached.

The study of the goodness of feature selection methods has been another interesting research area. Quantifying the sensitivity of the selected features to the variations in a training set is defined as the *stability* of the feature selection method. The performance of the feature importance of a method is influenced by the different subsamples of a training set. This, in turn, can affect the final selection of the features. The stability can be assessed by a pairwise comparison between the resulting subsets. The greater the similarity between the resulting subsets, the higher the stability. A number of measurements are available in the literature to quantify stability, and these are categorized into three representations: ranking, weighting, and indexing. Ranking- and weighting-based stability methods deal only with the full subset of features. In ranking, the correlations between the ranking vectors are measured, for example, by Spearman's Rank Correlation Coefficient. In weighting, Pearson's Correlation Coefficient method can be used to evaluate the weight of a full feature set. Unlike the ranking and weighting methods, stability by indexing methods considers the different cardinality of a feature subset. Examples of stability by index include Dice's Coefficient, Tanimoto Distance, Jaccard Index, and Kuncheva Index [99, 100]. In **Publication III**, we use dice coefficient to measure the similarity in the different feature subset. Method, such as random forest also ranks the importance of the features. Thus, it acts as an embedded feature-selection approach.

4 Accuracy assessment

The accuracy assessment is an important aspect in the model design process. It is motivated by two fundamental aims: a) the generalization ability of a learned model and b) the selection of the best one from a list of models. In general, the performance of a model is evaluated based on a fixed test set, which is independent of the training set. In the absence of a test set, the available data set is split into a fixed training and test set. However, if this test set only contains a small number of samples, this can increase the statistical uncertainty in estimating the test error. An alternative option is to repeat the training and testing procedures on different randomly chosen subsets of the available data. Of course, this will increase the computational cost as the model requires retraining for the different training subsets. The most common error estimation methods include cross-validation, resubstitution, bolstering, and bootstrap. A comparative discussion of these methods can be found in [101, 102].

Performance metrics are used to assess the quality of learned models and the preference for choosing a metric can be application specific. Sections 4.1 and 4.2 discuss useful metrics used in classification and regression problems, respectively. We consider cross-validation (CV) in Section 4.3. There are two main motivations for choosing CV in this thesis over other resampling methods. First, CV can be used to detect, for example, overfitting and to prevent it, a phenomenon where the model starts to learn noise from the data. The repetitive train-test split in CV reveals the performance of the model [103]. Second, we can apply CV as an alternative selection method for the hyperparameters. Section 4.4 discusses the pitfalls of CV. Another alternative approach, the Bayesian minimum mean square error estimator, can be used when CV is a poor choice for accuracy assessment. This section also introduces a novel accuracy metric that is more reliable and computationally faster than the alternative error estimators.

4.1 Performance measurement metrics in classification

In classification problems, the performance of a classifier can be described using a contingency table, called a *confusion matrix*. Table 4.1 illustrates a 2×2 confusion matrix for a binary classification problem, where the classes are usually given *positive* and *negative* class labels. Each entry represents the number of observations belonging to either true or false predictions for each of the two classes. There are four possible prediction outcomes. If an observation belonging to a true class is correctly predicted by the classifier, then the prediction is counted as *true positive (TP)*. Alternatively, if the observation is incorrectly predicted as a negative class, then the prediction is counted as *false negative (FN)*. On the other hand, if the observations belonging to the negative class are incorrectly predicted by the classifier as positive class, then the predictions are counted as *false positive (FP)*. Otherwise, if both the true and predicted class are

negative, then they are counted as *true negative (TN)*. The concept of the confusion matrix can also be extended to multiclass classification problems, where each entry (i, j) in the confusion matrix contains the number of observations belonging to class c_i but they are assigned to class c_j. Some important performance metrics which have been derived based on the confusion matrix are introduced below. More on the subject can be found in [104, 105].

Table 4.1: The 2×2 confusion matrix.

		Predicted class	
		positive	negative
True class	positive	TP	FN
	negative	FP	TN

Table 4.2 presents the definition of some common performance metrics based on the confusion matrix in Table 4.1. In order to evaluate a model in a classification task, the simplest and most standard metric is to measure the percentage of correctly predicted classifications. This is referred to as *accuracy* (ACC). In contrast, *error rate* (E) measures the percentage of misclassification errors. The main disadvantage of ACC and E is that they are only a single measure of performance and consider all correct and incorrect classifications with equal weight. The other measures in Table 4.2 give the ratio of correct classifications to each row or each column in the confusion matrix. Another popular metric, the F_1 score, which is widely used in information retrieval, considers both precision and recall such that $F_1 = 2 \cdot (\text{precision} \times \text{recall}/(\text{precision} + \text{recall}))$ [106].

Table 4.2: Example of common performance assessment metrics; P = TP + FN; N = FP + TN.

Accuracy (ACC)	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{P}+\mathrm{N}}$
Error rate (E)	$1 - \mathrm{ACC}$
True positive rate (TPR)	$\frac{\mathrm{TP}}{\mathrm{P}}$
False positive rate (FPR)	$\frac{\mathrm{FP}}{\mathrm{N}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Recall	$\frac{\mathrm{TP}}{\mathrm{P}}$
Sensitivity	$\frac{\mathrm{TP}}{\mathrm{P}}$
Specificity	$\frac{\mathrm{TN}}{\mathrm{N}}$

4.1.1 Receiver operating characteristic curve

The *Receiver Operating Characteristic* (ROC) curve is one of the simplest and most useful tools to represent the performance of a classifier graphically. The curve gives an idea of a trade-off between the number of TP (benefits) and FP (costs). The ROC curve can be viewed as a graph of TPR on the Y-axis versus FPR on the X-axis. This is analogous to a plot of the probability of detection versus the probability of false alarm in detection theory. In epidemiology, this curve plots the sensitivity against 1−specificity. The ROC curve was first introduced during World War II in military radar operation. The aim was to characterize the receiver operators' ability to correctly detect friendly or hostile aircrafts based on the signal on the radar screen. Subsequently, this tool was recognized for signal detection studies [107]. Later, the medical decision-making community adopted the use of the ROC curve for diagnostic testing [108]. In recent years, it has been used increasingly in machine-learning research [109].

In general, discrete classifiers produce a single decision, such as "0" or "1" and evaluate the classifiers with a test set by providing a single confusion matrix. This corresponds to one point in the ROC space. On the other hand, the probabilistic classifiers, such as logistic regression, yield a score or probability, the degree to which $\mathbf{x}$ is a member of class c_i. This score can be combined with a threshold T in order to produce a discrete classifier. As such, for binary classification, if $g(\mathbf{x}) > T$, then the response is $y = 1$, else $y = 0$. A common practice is to set $T = 0.5$. Nevertheless, an appropriate value for T depends on the domain knowledge. Theoretically, we can vary the value of T from $-\infty$ to ∞ and generate a curve in the ROC space. Each point in the curve represents a different T. However, computationally this is inadequate way to generate an ROC curve. Other techniques to estimate ROC curves are discussed in [110]. These includes nonparametric, parametric, and semi-parametric estimation methods.

As an example of a ROC space, Figure 4.1 illustrates a hypothetical ROC curve in a ROC space with seven scatter points representing seven different classifiers. These points have significant influence on the performance of the classifiers. The diagonal dash line in the figure divides the ROC space into lower and upper triangles. The points generated by the classifiers on this line (for instance, point A) have $TPR = FPR$ and they are making random guesses. These classifiers are ineffective and have no discriminative power to separate the classes. The points in the upper triangle represent classifiers that are better at separating the classes than the random guessing. Conversely, the points on the lower triangle represent classifiers that perform worse than random guessing. Thus, this lower triangle is usually empty. In Figure 4.1, point B performs much worse than point A. Note that the points in the lower triangle can be negated to produce points in the upper left triangle. Thus, the classifiers in the lower triangle have useful information, but they are applying the information incorrectly [111]. In the example, point B is in fact the negation of point C.

The classifiers represented by points in the left part of the ROC space are said to be "conservative" while the points generated by the classifiers in the right part of this space are said to be "liberal". The liberal classifiers can classify nearly all positives correctly but they produce high FPR. In contrast, the conservative classifiers have lower FPR, but produce high FN. Point D is more liberal than point C. An ideal classifier will generate point E, located at (0,1) with $FP = FN = 0$. The point F, located at (0,0) represents a classifier under which every instance is considered positively. In contrast, the point G, located at (1,1), represents a classifier that considers all instances to be negative.

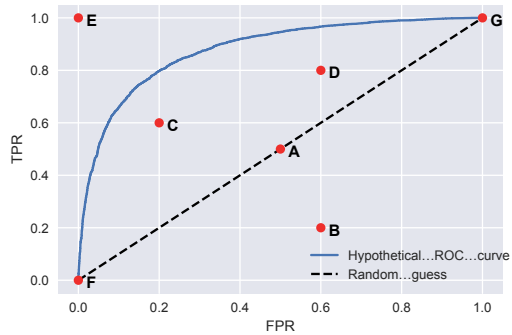


Figure 4.1: An example of a hypothetical ROC curve with seven different classifiers.

The standard ROC analysis for binary classification problem is also extended to multi-class problems, which is covered in the work of [112–114].

4.1.2 Area under the receiver operating characteristic curve

Often, a single number is desirable to express the performance of a classifier, instead of a graphical representation by the ROC curve. In a binary classification problem, we can express the ROC of any observation $\mathbf{x}$ for a threshold T as a function of TPR and FPR, such that $TPR(T) = \mathbb{P}\text{r}(\mathbb{P}\text{r}(c_1 \mid \mathbf{x}) > T \mid \mathbf{x} \in c_1)$ and $FPR(T) = \mathbb{P}\text{r}(\mathbb{P}\text{r}(c_1 \mid \mathbf{x}) > T \mid \mathbf{x} \in c_2)$, where $T \in [0, 1]$ [47]. Then, the ROC curve can be generated by considering all possible pairs of $\{TPR(T), FRP(T)\}$. The *area under the ROC curve* (AUROC) is, thus, the summary across all possible decision thresholds [115–118]. In general, the AUROC is estimated from the area under the ROC curve using quadrature [119].

The values of AUROC are between 0 and 1. However, the random guessing produces a diagonal line in the ROC space having an AUROC of 0.5, no rational classifiers have AUROC < 0.5. The AUROC can be computed using trapezoidal approximation. Different ROC estimates have different AUROC estimates. The AUROC is closely related to Mann-Whitney U and the Gini coefficient [120]. The AUROC is also extended to multiclass classification problem and can be solved either by micro- or macro-averaging [121].

4.2 Performance measurement metrics in regression

In regression problems, performance measurement metrics determine how close the prediction of a learned model is to the observed data. The most common metrics that are used in the literature include *mean absolute error* (MAE), *mean squared error* (MSE), and *coefficient of determination* (R^2 or R-squared) [122]. These metrics compare the difference between the estimated responses of the regression model and the true response. For instance, MAE describes the quantity of predictions that deviate from the true responses. It can be expressed as $\sum |y - \hat{y}|/n$, where y is the true response and $\hat{y}$ is the predicted response. In MAE, the interpretation is simple and straightforward. Moreover, positive and negative responses are tackled equally.

MSE is a quadratic version of MAE. We can express it as $\sum (y - \hat{y})^2/n$. This is also known as the Brier score when MSE is measured for probabilistic prediction [123]. The objective of a regression task is to minimize this MSE error. Another common choice for

error measure is the root mean squared error (RMSE). RMSE represents the square root of MSE. Both MAE and RMSE summarize the mean difference between y and $\hat{y}$ so, they are considered as good options for measuring a model's performance. However, RMSE is more sensitive to extreme error values than MAE. Another interesting error measure is median absolute error (MAD), which is robust to outliers. This metric considers the median of all absolute differences between y and $\hat{y}$.

Yet another popular performance measurement metric, is the coefficient of determination (R^2), which is mathematically expressed as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}. \tag{4.1}$$

Here, $\bar{y}$ is the average of the true response. The primary purpose of R^2 is to measure the goodness of the model in prediction. The value of R^2 is limited between 0 and 1. The 1 indicates regression predictions that perfectly fit the data. On the other hand, 0 indicates that the response cannot be predicted from the independent features. Any value of R^2 outside the limit indicates that something is wrong with the model. The disadvantage of R^2 is that the value of R^2 increases as we increase the number of features in the model. Thus, improving the R^2 does not always improve the performance of the model. This can be solved by using adjusted-R^2.

Consider an example with two models and a sequence of true responses, $y = \{2, 4, 6, 8\}$. The predicted output of the first model is, $\mathcal{M}_1 : \hat{y} = \{4, 6, 8, 10\}$. For the second model, the predicted output is, $\mathcal{M}_2 : \hat{y} = \{4, 6, 8, 12\}$. Table 4.3 summarizes the metrics for these two models.

Table 4.3: An example of a comparison of performance measurement metrics of two models.

Performance metric	$\mathcal{M}_1$	$\mathcal{M}_2$
R^2	0.2	−0.4
MAE	2.0	2.5
MSE	4.0	7
MAD	2.0	2.0

From the results, it is clear that the performance of the model $\mathcal{M}_1$ is better than that of the model $\mathcal{M}_2$. Except for MAD, the metrics lead to a similar conclusion. In fact, comparing the values of R^2, the evidence is quite strong. Here, MAD tells nothing about these models. In this thesis, we use MAE, MSE and R^2 metrics to provide us with comprehensive information on the regression models.

4.3 Cross-validation

A model is learned on the available data set. The purpose of using cross-validation is to estimate how well a learned model will perform, i.e., to get an insight about the generalization ability of the learned model for an independent data set. In data-rich settings, the *holdout estimate* is used to partition the data into two mutually exclusive parts. One part is used as the training data to train the model, whilst the other part is used as the test or validation data to evaluate the model's performance (see Figure 2.2 as an example). In general, the data split is $80\% - 20\%$. This approach increases the accuracy of the model if more data are added in the training set. In fact, the error

estimated from the holdout test set can be unbiased when $n \to \infty$, and the variance of the holdout estimator is approximately equal to the variance of the true error [124]. However, if the data are insufficient, then splitting the data in such a conventional approach may suffer from either poor design choice of the classifier with fewer data points or poor error estimation [124, 125].

The K-fold cross-validation (K-fold CV) is a popular approach to enhance the model performance in limited-data settings. The data are divided into approximately k equal-sized partitions or folds, as illustrated in Figure 4.2, and the procedure is repeated k times. At each iteration, one of the k folds (white regions) is used as test or validation data and the remaining $(k-1)$ folds (gray regions) are used for training the model. The process is repeated until all the k folds have been used as test data to measure the model performance. The k results from the folds are then averaged to compute a single estimate. The main advantage of this approach is that all the data are used for both training and testing, and each data point is used exactly once for testing. However, the disadvantage of this approach is that the model is retrained k times, so the training requires more computation.

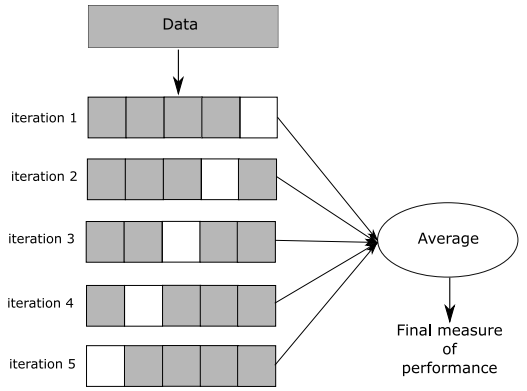


Figure 4.2: Example of 5-fold cross-validation.

In general, the preferred values of K are either 5 or 10. An extreme choice of $K = n$ is known as *leave-one-out* cross-validation (LOOCV). In this case, the model is trained n times and only one data point is used for testing at each iteration. As such, this approach is computationally expensive. With a large number of folds, the bias of the true error rate estimator will be small. However, the variance of the true error rate will be large. In contrast, a lower number of folds has higher bias but lower variance. Hence, LOOCV has low bias but can have high variance.

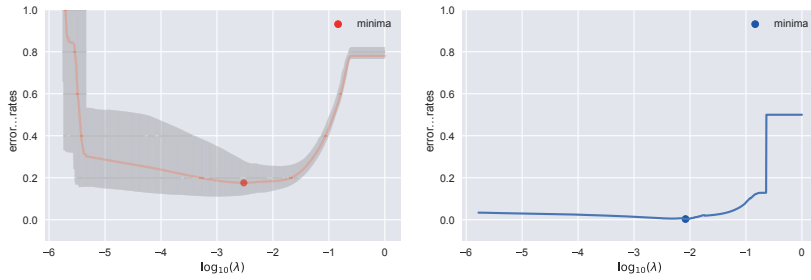
In classification, instead of splitting the data by random sampling, we can use *stratified* sampling. In this case, the data are first separated according to the class labels, also referred to as *stratum*. Then, a subset of data is chosen from each stratum with a probability. The advantage of stratified sampling over the traditional random sampling is to minimize the selection bias and to ensure selected samples from all classes are balanced. Therefore, stratified sampling can overcome the challenges of an imbalanced data set, where data from one class are either under-represented or over-represented [126].

4.4 Bayesian approach to accuracy assessment

In the model design process, model selection is a critical phase as most machine-learning models require adjusting a set of hyperparameters, which can have an effect on the model's performance. For instance, the aforementioned linear classifiers depends on the regularization hyperparameter λ. The neural networks are sensitive to a number of hyperparameters, such as the number of hidden units, learning rate, dropout rate and so on [77].

In order to find an optimal value for the hyperparameter, the simplest and most traditional approach is grid search, which is an exhaustive searching approach. The grid search is performed through manually specified values of the hyperparameter space and the aim is to optimize a performance metric, often using a cross-validated training set. As an example, Figure 4.3 (a) illustrates the error curves generated along a regularization path of a model-selection hyperparameter λ with ranges from 10^{-9} to 1. The error rates are estimated for 5-fold cross-validation. The procedure is repeated 100 times and the red error curve shown in Figure 4.3 (a) is the averaged error estimate.

The randomness in cross-validation during split operations introduces deviations in the error estimate from one iteration to another as seen in Figure 4.3 (a). Moreover, the splitting of the data for K-fold cross-validation with a very small number of samples, for example 20 (see **Publication III**), is inappropriate as there are not enough samples in each fold. Although LOOCV is another alternative, this will lead to high variance in the error estimate. In addition to these issues, K-fold cross-validation can be computationally expensive as the model is learned K times for K folds and one more for the whole data set.



(a) 5-fold cross-validation **(b)** Bayesian error estimation

Figure 4.3: Estimation of error rates for different values of model-selection hyperparameter λ. a) The gray curves represent the error estimates for different iterations in 5-fold cross-validation. Due to the randomness in the cross-validation split operation, the error estimates are deviated from one iteration to another. The red curve is the average of the all the gray curves. The minimum error rate is shown in the red circle. b) The Bayesian error estimation has only one blue error curve since the method does not require any additional split of the training data set. The minimum error rate is shown in the blue circle. The image is regenerated from **Publication III**.

4.4.1 Bayesian error estimation

An alternative error estimation approach, the *Bayesian minimum mean square error estimator* (BEE) has recently been introduced for binary classification problems using discrete [127] and linear [128] classifiers. In this approach, the classification errors are measured directly from the training set. This significantly improves the speed of computation during the model-selection process. Moreover, both the accuracy and the stability of the learned model have shown to be improved compared to the other error estimation approaches, such as resubstitution, bootstrapping, and CV [129, 130]. The error estimate of BEE in Figure 4.3 (b) is a single deterministic error curve.

Consider a two-class classification problem with labels $c \in \{1, 2\}$. We assume the samples from each class are independent and identically distributed Gaussian random variables. Let us denote the distribution parameter for class c as $\mathbf{\Theta}_c$. Given a prior probability $\Pr(\mathbf{\Theta}_c)$, we can write the posterior probability density function of the parameter for class $c \in \{1, 2\}$ using the Bayes rule,

$$\Pr^*(\mathbf{\Theta}_c \mid \mathbf{X}, \mathbf{y}) \propto \Pr(\mathbf{\Theta}_c) \prod_{i:y_i=c} \Pr(\mathbf{x}_i \mid \mathbf{\Theta}_c), \tag{4.2}$$

where $\Pr(\mathbf{x}_i \mid \mathbf{\Theta}_c)$ are the Gaussian class conditional densities for $c \in \{1, 2\}$. BEE is a minimum mean squared estimator (MMSE) that minimizes the expectation between the estimated and the true classification error. BEE is defined by [128] for a linear classifier as

$$\text{BEE} \triangleq \mathbb{E}[\varepsilon \mid \mathbf{X}, \mathbf{y}] = \Pr(c = 1)\, \mathbb{E}[\varepsilon_1 \mid \mathbf{X}, \mathbf{y}] + \Pr(c = 2)\, \mathbb{E}[\varepsilon_2 \mid \mathbf{X}, \mathbf{y}], \tag{4.3}$$

with the expected classification error of samples from class c given by

$$\mathbb{E}[\varepsilon_c \mid \mathbf{X}, \mathbf{y}] = \int \varepsilon_c(\mathbf{\Theta}_c)\, \Pr^*(\mathbf{\Theta}_c \mid \mathbf{X}, \mathbf{y})\, \mathrm{d}\mathbf{\Theta}_c, \tag{4.4}$$

where $\varepsilon_c(\mathbf{\Theta}_c)$ denotes the true classification error for class c. We define the distribution parameter $\mathbf{\Theta}_c = \{\boldsymbol{\mu}_c, \mathbf{\Sigma}_c\}$, where $\boldsymbol{\mu}_c \in \mathbb{R}^p$ and $\mathbf{\Sigma}_c \in \mathbb{R}^{p \times p}$ are the center and the invertible covariance matrix of the Gaussian class c, respectively. For invertible $\mathbf{\Sigma}_c$, we assume the priors are of the form

$$\Pr(\mathbf{\Theta}_c) = \Pr(\boldsymbol{\mu}_c \mid \mathbf{\Sigma}_c)\, \Pr(\mathbf{\Sigma}_c),$$

where

$$\Pr(\boldsymbol{\mu}_c \mid \mathbf{\Sigma}_c) \sim \mathcal{N}\left(\mathbf{m}, \frac{\mathbf{\Sigma}_c}{\nu}\right)$$

and

$$\Pr(\mathbf{\Sigma}_c) \sim \mathcal{W}^{-1}(\mathbf{S}, \kappa).$$

That is, the mean of class c conditioned on the covariance is Gaussian with mean $\mathbf{m}$ and covariance $\mathbf{\Sigma}_c/\nu$, and the marginal distribution of covariance is an Inverse Wishart distribution with parameters $\mathbf{S}$ and κ. Here, $\nu \in \mathbb{R}, \kappa \in \mathbb{R}, \mathbf{S} \in \mathbb{R}^{P \times P}$ and $\mathbf{m} \in \mathbb{R}^P$

are the hyperparameters of the Bayesian model. Details of these hyperparameters are discussed in [128].

For a linear classifier of the form $g(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\beta} + \beta_0$ and with fixed distribution parameters, the true classification error can be given by

$$\varepsilon_c(\boldsymbol{\Theta}_c) = \Phi\left(\frac{(-1)^c g(\boldsymbol{\mu}_c)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}_c\boldsymbol{\beta}}}\right), \tag{4.5}$$

where $\Phi(.)$ is the of unit normal Gaussian cumulative distribution function. Now, the closed form solution of BEE in the Gaussian model with general and scaled covariance matrices can be specified respectively as,

$$\mathbb{E}[\varepsilon \mid \mathbf{X}, \mathbf{y}] = \frac{1}{2}\left(1 + \frac{\text{sgn}(A)}{2} I\left(\frac{A^2}{A^2 + \boldsymbol{\beta}^T\mathbf{S}^*\boldsymbol{\beta}}; \frac{1}{2}, \frac{\kappa^* - p + 1}{2}\right)\right), \tag{4.6}$$

and

$$\mathbb{E}[\varepsilon \mid \mathbf{X}, \mathbf{y}] = \frac{1}{2}\left(1 + \text{sgn}(A)\, I\left(\frac{A^2}{A^2 + 2\beta}; \frac{1}{2}, \alpha\right)\right). \tag{4.7}$$

Here,

$$\begin{aligned} A &= (-1)^c g(\mathbf{m}^*)\sqrt{\frac{\nu^*}{\nu^* + 1}} \\ \alpha &= \frac{(\kappa^* + p + 1)p}{2} - 1 \\ \beta &= \frac{1}{2}\,\text{trace}(S^*), \end{aligned} \tag{4.8}$$

and $I(x; \alpha, \beta)$ is an incomplete beta function. The proofs of Eq. 4.6 and Eq. 4.7 are covered in [128]. This Bayesian framework can be used to measure alternative accuracy measures, such as the receiver operating characteristic curve (ROC) which is discussed in the next section.

4.4.2 Bayesian receiver operating characteristics curve

The Bayesian estimation of ROC has been studied in [131] by deriving optimal estimators for the true positive rate (TPR) and false positive rate (FPR), and iteratively sampling the TPR-FPR space. Then, we can measure the area under the ROC curve (AUROC). We call this approach *Empirical Bayesian AUROC* (EBAUROC). If we modify the parameters with a fixed common covariance matrix $\boldsymbol{\Sigma}$ and unknown means for the binary linear classification problem i.e., $\boldsymbol{\Theta}_c = \{\boldsymbol{\mu}_c, \boldsymbol{\Sigma}\}$, then we can derive a closed-form solution for computing AUROC in the Bayesian framework, which can directly compute the ROC instead of classification error. The population-AUROC for a linear classifier with known distribution parameters can be defined by [132]

$$\mathrm{AUROC}(\boldsymbol{\beta} \mid \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \Phi\left(\frac{\boldsymbol{\beta}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\right). \tag{4.9}$$

Here, Φ is the Gaussian (cumulative) distribution function. Now, we can define the closed-form Bayesian AUROC (CBAUROC) as follows:

Definition 1. *The* Bayesian Area Under the Receiver Operating Characteristic Curve *of a linear classifier with coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ classifying p-dimensional samples from two Gaussian distributions with parameters $\boldsymbol{\Theta}_1 = \{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}\}$ and $\boldsymbol{\Theta}_2 = \{\boldsymbol{\mu}_2, \boldsymbol{\Sigma}\}$ is the posterior expectation of the AUROC in Eq. 4.9:*

$$\mathrm{CBAUROC}(\boldsymbol{\beta}) \quad = \int \mathrm{AUROC}(\boldsymbol{\beta} \mid \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)\, \mathbb{P}\mathrm{r}^*(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2)\, \mathrm{d}\boldsymbol{\Theta}_1 \mathrm{d}\boldsymbol{\Theta}_2. \tag{4.10}$$

For fixed κ, $\mathbf{S}$, ν and $\mathbf{m}$, the posterior probabilities of the distribution parameters are given by

$$\mathbb{P}\mathrm{r}^*(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \propto \mathbb{P}\mathrm{r}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \prod_{y_c=1} f_{\boldsymbol{\Theta}_1}(\mathbf{x}^c) \prod_{y_c=2} f_{\boldsymbol{\Theta}_2}(\mathbf{x}^c). \tag{4.11}$$

We assume that $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are independent given $\boldsymbol{\Sigma}$. Then, the prior is

$$\mathbb{P}\mathrm{r}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \mathbb{P}\mathrm{r}(\boldsymbol{\Sigma}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \mathbb{P}\mathrm{r}(\boldsymbol{\Sigma})\, \mathbb{P}\mathrm{r}(\boldsymbol{\mu}_1 \mid \boldsymbol{\Sigma})\, \mathbb{P}\mathrm{r}(\boldsymbol{\mu}_2 \mid \boldsymbol{\Sigma}) \tag{4.12}$$

Since $\mathbb{P}\mathrm{r}(\boldsymbol{\Sigma}) \sim \mathcal{W}^{-1}(\mathbf{S}, \kappa)$ and $\mathbb{P}\mathrm{r}(\boldsymbol{\mu}_c \mid \boldsymbol{\Sigma}) \sim \mathcal{N}(\mathbf{m}, \frac{\boldsymbol{\Sigma}}{\nu})$, we can write Eq. 4.12 as

$$\begin{aligned}
\mathbb{P}\mathrm{r}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \propto\ & \det(\boldsymbol{\Sigma})^{-(\kappa+p+1)/2} \exp\left(-\frac{1}{2}\operatorname{trace}(\mathbf{S}\boldsymbol{\Sigma}^{-1})\right) \\
& \times \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{\nu_1}{2}(\boldsymbol{\mu}_1 - \mathbf{m}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \mathbf{m}_1)\right) \\
& \times \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{\nu_2}{2}(\boldsymbol{\mu}_2 - \mathbf{m}_2)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \mathbf{m}_2)\right).
\end{aligned}$$

The sample means and covariances are given by

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{n_c}\sum_{i=1}^{n_c} \mathbf{x}_i^c \text{ and } \hat{\boldsymbol{\Sigma}}_c = \frac{1}{n_c - 1}\sum_{i=1}^{n_c} (\mathbf{x}_i^c - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i^c - \hat{\boldsymbol{\mu}}_c)^T$$

Now, the posterior distribution in Eq. 4.11 can be rewritten with simplification as

$$\begin{aligned}
\mathbb{P}\mathrm{r}^*(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) \propto\ & \det(\boldsymbol{\Sigma})^{-(\kappa^*+p+1)/2} \exp\left(-\frac{1}{2}\operatorname{trace}\left(\mathbf{S}^*\boldsymbol{\Sigma}^{-1}\right)\right) \\
& \times \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{\nu_1^*}{2}(\boldsymbol{\mu}_1 - \mathbf{m}_1^*)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \mathbf{m}_1^*)\right) \\
& \times \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{\nu_2^*}{2}(\boldsymbol{\mu}_2 - \mathbf{m}_2^*)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \mathbf{m}_2^*)\right) \\
& \propto \mathbb{P}\mathrm{r}^*(\boldsymbol{\Sigma})\, \mathbb{P}\mathrm{r}^*(\boldsymbol{\mu}_1 \mid \boldsymbol{\Sigma})\, \mathbb{P}\mathrm{r}^*(\boldsymbol{\mu}_2 \mid \boldsymbol{\Sigma}).
\end{aligned} \tag{4.13}$$

The updated hyperparameters are

$$\begin{aligned}
\kappa^* &= \kappa + n_1 + n_2 = \kappa + n,\\
\nu_1^* &= \nu_1 + n_1,\\
\nu_2^* &= \nu_2 + n_2,\\
\mathbf{S}^* &= (n_1 - 1)\hat{\mathbf{\Sigma}}_1 + (n_2 - 1)\hat{\mathbf{\Sigma}}_2 + \mathbf{S}\\
&\quad + \frac{n_1\nu_1}{n_1 + \nu_1}(\hat{\boldsymbol{\mu}}_1 - \mathbf{m}_1)(\hat{\boldsymbol{\mu}}_1 - \mathbf{m}_1)^T\\
&\quad + \frac{n_2\nu_2}{n_2 + \nu_2}(\hat{\boldsymbol{\mu}}_2 - \mathbf{m}_2)(\hat{\boldsymbol{\mu}}_2 - \mathbf{m}_2)^T,\\
\mathbf{m}_1^* &= \frac{(n_1\hat{\boldsymbol{\mu}}_1 + \nu_1\mathbf{m}_1)}{(n_1 + \nu_1)}\\
\mathbf{m}_2^* &= \frac{(n_2\hat{\boldsymbol{\mu}}_2 + \nu_2\mathbf{m}_2)}{(n_2 + \nu_2)}.
\end{aligned} \tag{4.14}$$

We can rewrite Eq. (4.10) with the notation of Eq. (4.13) such that

$$\begin{aligned}
\mathrm{CBAUROC}(\boldsymbol{\beta}) = \mathbb{E}[\mathrm{AUROC} \mid \boldsymbol{\beta}] = \int\int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \mathrm{AUROC}(\boldsymbol{\beta})\\
\times \mathbb{P}\mathrm{r}^*(\boldsymbol{\mu}_1|\mathbf{\Sigma})\,\mathbb{P}\mathrm{r}^*(\boldsymbol{\mu}_2|\mathbf{\Sigma})\,\mathbb{P}\mathrm{r}^*(\mathbf{\Sigma})\,\mathrm{d}\boldsymbol{\mu}_1\,\mathrm{d}\boldsymbol{\mu}_2\mathrm{d}\mathbf{\Sigma}.
\end{aligned} \tag{4.15}$$

In order to evaluate the two inner integrals, we state the following lemma.

Lemma 1. *Assuming a fixed common covariance matrix* $\mathbf{\Sigma}$ *for the classes, the expected posterior* AUROC *over class means* $\boldsymbol{\mu}_1$ *and* $\boldsymbol{\mu}_2$ *is given by*

$$\begin{aligned}
&\int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \mathrm{AUROC}(\boldsymbol{\beta})\,\mathbb{P}\mathrm{r}^*(\boldsymbol{\mu}_1 \mid \mathbf{\Sigma})\,\mathbb{P}\mathrm{r}^*(\boldsymbol{\mu}_2 \mid \mathbf{\Sigma})\,\mathrm{d}\boldsymbol{\mu}_1\,\mathrm{d}\boldsymbol{\mu}_2\\
&= \int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \Phi\left(\frac{\boldsymbol{\beta}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)}{\sqrt{2\boldsymbol{\beta}^T\mathbf{\Sigma}\boldsymbol{\beta}}}\right) f_{\mathbf{\Theta}_1^*}(\boldsymbol{\mu}_1)\, f_{\mathbf{\Theta}_2^*}(\boldsymbol{\mu}_2)\mathrm{d}\boldsymbol{\mu}_1\mathrm{d}\boldsymbol{\mu}_2\\
&= \Phi\left(\frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\mathbf{\Sigma}\boldsymbol{\beta}}}\sqrt{\frac{2\nu_1^*\nu_2^*}{\nu_1^* + \nu_2^* + 2\nu_1^*\nu_2^*}}\right).
\end{aligned} \tag{4.16}$$

Proof. Let

$$M = \int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \Phi\left(\frac{\boldsymbol{\beta}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)}{\sqrt{2\boldsymbol{\beta}^T\mathbf{\Sigma}\boldsymbol{\beta}}}\right) f_{\mathbf{\Theta}_1^*}(\boldsymbol{\mu}_1)\, f_{\mathbf{\Theta}_2^*}(\boldsymbol{\mu}_2)\,\mathrm{d}\boldsymbol{\mu}_1\,\mathrm{d}\boldsymbol{\mu}_2. \tag{4.17}$$

We know, given a fixed covariance matrix, the posterior density for the mean is Gaussian. Thus, we can write,

$$f_{\mathbf{\Theta}_c^*}(\boldsymbol{\mu}_c) = \frac{{\nu_c^*}^{\frac{p}{2}}}{(2\pi)^{\frac{p}{2}}\det(\mathbf{\Sigma})^{\frac{1}{2}}}\exp\left(-\frac{\nu_c^*}{2}(\boldsymbol{\mu}_c - \mathbf{m}_c^*)^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_c - \mathbf{m}_c^*)\right).$$

Replacing these expressions in Eq. 4.17, we get

$$\begin{aligned} M = \int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \Phi\left(\frac{\boldsymbol{\beta}^T(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\right) \exp\left(-\frac{\nu_1^*}{2}(\boldsymbol{\mu}_1-\mathbf{m}_1^*)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\mathbf{m}_1^*)\right) \\ \times \exp\left(-\frac{\nu_2^*}{2}(\boldsymbol{\mu}_2-\mathbf{m}_2^*)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2-\mathbf{m}_2^*)\right) \frac{{\nu_1^*}^{\frac{p}{2}}{\nu_2^*}^{\frac{p}{2}}}{(2\pi)^p\det(\boldsymbol{\Sigma})}\,\mathrm{d}\boldsymbol{\mu}_1\,\mathrm{d}\boldsymbol{\mu}_2. \end{aligned} \tag{4.18}$$

Since $\boldsymbol{\Sigma}$ is an invertible covariance matrix and by singular value decomposition, we can say, $\boldsymbol{\Sigma} = \mathbf{U}\mathbf{U}^T$ with $\det(\boldsymbol{\Sigma}) = \det(\mathbf{U})^2$. Moreover, making changes of variables $\mathbf{z}_c = \sqrt{\nu_c^*}\mathbf{U}^{-1}(\boldsymbol{\mu}_c - \mathbf{m}_c^*)$ we obtain,

$$\begin{aligned} M &= \int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \Phi\left(\frac{\boldsymbol{\beta}^T(\frac{1}{\sqrt{\nu_2^*}}\mathbf{z}_2\mathbf{U}+\mathbf{m}_2^*-\frac{1}{\sqrt{\nu_1^*}}\mathbf{z}_1\mathbf{U}+\mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\right) \\ &\times \frac{1}{(2\pi)^p}\exp\left(-\frac{\mathbf{z}_1^T\mathbf{z}_1+\mathbf{z}_2^T\mathbf{z}_2}{2}\right)\mathrm{d}\mathbf{z}_1\,\mathrm{d}\mathbf{z}_2 \\ &= \int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \Phi\left(\frac{\frac{1}{\sqrt{\nu_2^*}}\boldsymbol{\beta}^T\mathbf{U}\mathbf{z}_2+\boldsymbol{\beta}^T\mathbf{m}_2^*-\frac{1}{\sqrt{\nu_1^*}}\boldsymbol{\beta}^T\mathbf{U}\mathbf{z}_1-\boldsymbol{\beta}^T\mathbf{m}_1^*}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\right) \\ &\times \frac{1}{(2\pi)^p}\exp\left(-\frac{\mathbf{z}_1^T\mathbf{z}_1+\mathbf{z}_2^T\mathbf{z}_2}{2}\right)\mathrm{d}\mathbf{z}_1\,\mathrm{d}\mathbf{z}_2. \end{aligned} \tag{4.19}$$

Let us define

$$\begin{aligned} \mathbf{a}_c &= \frac{\mathbf{U}^T\boldsymbol{\beta}}{\sqrt{\nu_c^*}\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}, \\ b_c &= \frac{\boldsymbol{\beta}^T\mathbf{m}_c^*}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}, \\ \|\mathbf{a}_c\|^2 &= \frac{1}{2\nu_c^*} \end{aligned} \tag{4.20}$$

to obtain

$$\begin{aligned} M &= \int_{\mathbb{R}^p}\int_{\mathbb{R}^p} \Phi\left([\mathbf{a}_2,-\mathbf{a}_1][\mathbf{z}_2,\mathbf{z}_1]^T+(b_2-b_1)\right)\frac{1}{(2\pi)^p}\exp\left(-\frac{[\mathbf{z}_1,\mathbf{z}_2][\mathbf{z}_1,\mathbf{z}_2]^T}{2}\right)\mathrm{d}\mathbf{z}_1\mathrm{d}\mathbf{z}_2 \\ &= \int_{\mathbb{R}^p}\int_{\mathbb{R}^p}\int_{R(x)} \frac{1}{(2\pi)^{p+\frac{1}{2}}}\exp\left(-\frac{x^2+[\mathbf{z}_1,\mathbf{z}_2][\mathbf{z}_1,\mathbf{z}_2]^T}{2}\right)\mathrm{d}x\,\mathrm{d}\mathbf{z}_1\,\mathrm{d}\mathbf{z}_2, \end{aligned} \tag{4.21}$$

where $R(x) = \{x | x < [\mathbf{a}_2,-\mathbf{a}_1][\mathbf{z}_2,\mathbf{z}_1]^T + (b_2-b_1)\}$. Analogously to [128, Appendix B] we conclude that Eq. 4.21 gives the error of the $2p+1$ dimensional two-class classifier $\bar{g}([\mathbf{z}_1,\mathbf{z}_2]^T, x) = [\mathbf{a}_2,-\mathbf{a}_1][\mathbf{z}_2,\mathbf{z}_1]^T - x + (b_2-b_1)$ originating from the class c_1 Gaussian distribution with zero mean and identity covariance. Thus, from [128, Eq. 9] it follows

$$
\begin{aligned}
M &= \Phi\left(\frac{b_2 - b_1}{\sqrt{\|[\mathbf{a}_2, -\mathbf{a}_1]^T\|^2 + 1}}\right) \\
&= \Phi\left(\frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}\sqrt{\frac{1}{2\nu_2^*} + \frac{1}{2\nu_1^*} + 1}}\right) \\
&= \Phi\left(\frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\sqrt{\frac{2\nu_1^*\nu_2^*}{\nu_1^* + \nu_2^* + 2\nu_1^*\nu_2^*}}\right).
\end{aligned}
$$

Thus, we have concluded the proof. □

Substitution of the above result into Eq. (4.15) yields

$$
\mathrm{CBAUROC}(\boldsymbol{\beta}) = \int \Phi\left(\frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\sqrt{\frac{2\nu_1^*\nu_2^*}{\nu_1^* + \nu_2^* + 2\nu_1^*\nu_2^*}}\right) \mathbb{P}\mathrm{r}^*(\boldsymbol{\Sigma})\,\mathrm{d}\boldsymbol{\Sigma}. \tag{4.22}
$$

Next, we will state the main proposition of **Publication V** by solving this integral.

4.4.3 Normal-inverse Wishart distribution for covariance

Proposition 1. *Let $\psi(\cdot) : \mathbb{R}^p \mapsto \{1, 2\}$ denote a binary linear classifier parameterized by coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ and intercept $\beta_0 \in \mathbb{R}$. The Bayesian Area Under Curve estimate given training samples $\mathfrak{D}_{\mathtt{train}}$ assuming a common inverse Wishart distributed covariance for both classes is then given as*

$$
\mathrm{CBAUROC}(\boldsymbol{\beta}) = \frac{1}{2} + \frac{\mathrm{sgn}(A^*)}{2} I\left(\frac{A^{*2}}{A^{*2} + \boldsymbol{\beta}^T\mathbf{S}^*\boldsymbol{\beta}}; \frac{1}{2}, \frac{\kappa^* - p + 1}{2}\right). \tag{4.23}
$$

with

$$
A^* = \frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)\sqrt{\nu_1^*\nu_2^*}}{\sqrt{\nu_1^* + \nu_1^* + 2\nu_1^*\nu_2^*}},
$$

and ν_1^*, ν_2^*, $\mathbf{m}_1^*$, $\mathbf{m}_2^*$, $\mathbf{S}^*$ and κ^* as in Eq. 4.14.

Proof. The posterior density $\mathbb{P}\mathrm{r}^*(\boldsymbol{\Sigma})$ has an inverse Wishart distribution, *i.e.*,

$$
\mathbb{P}\mathrm{r}^*(\boldsymbol{\Sigma}) = \mathcal{W}^{-1}(\boldsymbol{\Sigma}; \mathbf{S}^*, \kappa^*)
$$

with hyperparameters $\mathbf{S}^*$ and κ^*. If we take

$$
A^* = \frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)\sqrt{\nu_1^*\nu_2^*}}{\sqrt{\nu_1^* + \nu_1^* + 2\nu_1^*\nu_2^*}},
$$

then Eq. 4.22 gets the form

$$\mathrm{CBAUROC}(\boldsymbol{\beta}) = \int_{R(\boldsymbol{\Sigma})} \Phi\left(\frac{A^*}{\sqrt{\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\right) \mathbb{P}\mathrm{r}^*(\boldsymbol{\Sigma})\,\mathrm{d}\boldsymbol{\Sigma}, \tag{4.24}$$

where $R(\boldsymbol{\Sigma}) = \{\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p} \mid \boldsymbol{\Sigma} \text{ positive definite}\}$. By applying [128, Lemma E.1], we arrive at

$$\int_{R(\boldsymbol{\Sigma})} \Phi\left(\frac{A^*}{\sqrt{\boldsymbol{\beta}^T\boldsymbol{\Sigma}\boldsymbol{\beta}}}\right) \mathbb{P}\mathrm{r}^*(\boldsymbol{\Sigma})\mathrm{d}\boldsymbol{\Sigma} = \frac{1}{2} + \frac{\mathrm{sgn}(A^*)}{2} I\left(\frac{A^{*2}}{A^{*2} + \boldsymbol{\beta}^T\mathbf{S}^*\boldsymbol{\beta}}; \frac{1}{2}, \frac{\kappa^* - p + 1}{2}\right). \tag{4.25}$$

This concludes the proof. □

4.4.4 Inverse Gamma distribution for covariance

Proposition 2. *Let $\psi(\cdot) : \mathbb{R}^p \mapsto \{1, 2\}$ denote a binary linear classifier parameterized by coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$ and intercept $\beta_0 \in \mathbb{R}$. The Bayesian Area Under Curve estimate given training samples $\mathfrak{D}_{\mathtt{train}}$ assuming scaled identity covariance $\boldsymbol{\Sigma} = \sigma\mathbf{I}$ for both classes is then given as*

$$\mathrm{CBAUROC}(\boldsymbol{\beta}) = \frac{1}{2}\left(1 + \mathrm{sgn}(A^*) I\left(\frac{A^{*2}}{A^{*2} + 2\beta}; \frac{1}{2}, \alpha\right)\right) \tag{4.26}$$

with

$$\alpha = \frac{((\kappa^* + p + 1)p)}{2} - 1, \qquad \beta = \frac{1}{2}\,\mathrm{trace}(\mathbf{S}^*),$$

and

$$A^* = \frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\beta}}} \frac{\sqrt{2\nu_1^*\nu_2^*}}{\sqrt{\nu_1^* + \nu_1^* + 2\nu_1^*\nu_2^*}}.$$

Proof. Since $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ is a scaled identity matrix, the posterior density for the scalar multiplier $\mathbb{P}\mathrm{r}^*(\sigma^2)$ has an inverse gamma distribution, *i.e.*,

$$\mathbb{P}\mathrm{r}^*(\sigma^2) = \Gamma^{-1}(\sigma^2; \alpha, \beta)$$

with the hyperparameters $\alpha = \frac{((\kappa^*+p+1)p)}{2} - 1$ and $\beta = \frac{1}{2}\,\mathrm{trace}(\mathbf{S}^*)$. If we take

$$A^* = \frac{\boldsymbol{\beta}^T(\mathbf{m}_2^* - \mathbf{m}_1^*)}{\sqrt{2\boldsymbol{\beta}^T\boldsymbol{\beta}}} \frac{\sqrt{2\nu_1^*\nu_2^*}}{\sqrt{\nu_1^* + \nu_1^* + 2\nu_1^*\nu_2^*}},$$

then Eq. 4.22 gets the form

$$\mathrm{CBAUROC} = \int_{\sigma^2>0} \Phi\left(\frac{A^*}{\sqrt{\sigma^2}}\right) \mathbb{P}\mathrm{r}^*(\sigma^2)\,\mathrm{d}\sigma^2. \tag{4.27}$$

By applying [128, Lemma D.1], we arrive at

$$\int_{\sigma^2>0} \Phi\left(\frac{A^*}{\sqrt{\sigma^2}}\right) \mathbb{P}\mathrm{r}^*(\sigma^2)\,\mathrm{d}\sigma^2 = \frac{1}{2}\left(1 + \mathrm{sgn}(A^*) I\left(\frac{A^{*2}}{A^{*2} + 2\beta}; \frac{1}{2}, \alpha\right)\right). \tag{4.28}$$

This concludes the proof. □

5 Applications in biology

In recent years, the rapid developments in computational biology and bioinformatics have generated an enormous amount of biological data. These data require sophisticated computational tools to interpret various biological systems and their relationships. Thus, the integration of applications and the development of data analytical methods have become an active research area to solve biological problems. These include, for example, protein structure prediction, gene classification, cancer classification, and computational modeling of cell biological processes. Some of these applications are introduced in this chapter. The utilization of various machine-learning algorithms for these applications is based on publications that the author has been involved in.

First, Section 5.1 describes the application areas of **Publications I − V** and the associated challenges. Then, Section 5.2 summarizes our main results in solving those challenges. These results are the key findings associated with the publications and reflect the objectives listed in Section 1.2. Finally, the chapter concludes with a discussion of these findings.

5.1 Case studies

The application areas in this thesis include the need for data analysis in bioprocess development, to study the behavior of foodborne pathogens in bacteria at different temperatures, to explore the pathways that regulate the cellulose production in microorganisms, and to explore the feature selection ability in cancer-related research. We will provide an overview for each of these areas in the following.

5.1.1 Bioprocess data mining

Life sciences with biotechnology have improved our living standards and enabled the industrial production of many useful everyday products. Biotechnology exploits processes that use biological systems or living organisms (such as enzymes, yeast or other microorganisms) to produce and develop products according to human needs. Traditional biotechnology started with the fermentation process. One of the earliest such process was reported in 6000 B.C., where grapes were fermented by Neolithic cultures to produce wine. During the same period, Sumerians and Babylonians used microbial yeast to produce beer [133]. Gradually, the knowledge of fermentation was expanded to produce foods (such as cheese, yogurt, and vinegar) and to preserve other food products. However, the process was not fully understood until 1857 when Louis Pasteur discovered that microbial activity was the basis of fermentation in alcohol [134]. Further, the discovery of penicillin from a mold by Alexander Fleming in 1928, opened another paradigm of biotechnology in medicine and drug development. Later, large-scale fermentation techniques were adopted to produce penicillin commercially in large quantities [135, 136]. Modern biotechnology

began its journey in the late 20th century and has expanded into genomics, proteomics, genetic engineering, biochemistry, molecular biology, pharmaceutical therapies, and many other disciplines [137].

In recent years, major industrial application areas of biotechnology include medicine, agriculture, biofuels, and environmental uses. These industries have designed and developed sophisticated bioprocesses for utilizing living organisms to produce cost-efficient, environment-friendly, and high-quality products. Designing these bioprocesses involves maximization of yields as well as identifying an optimal environment, which requires a substantial amount of experiments in order to produce a targeted biomolecule. The experiments can be, for example, finding the right concentration of nutrients (such as carbon, oxygen, nitrogen, phosphorous, sulfur, minerals) for a fermentation culture; eliminating toxic components (such as carbon dioxide); controlling the behavior of process parameters (such as temperature or the acidity level of an aqueous solution) [138].

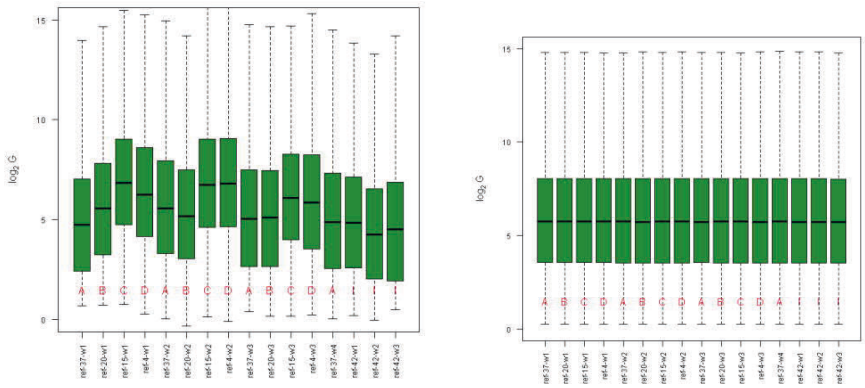
In addition to designing bioprocesses, industrial biotechnology is also generating data for process monitoring, controlling, and boosting the manufacturing performance with the aid of advanced sensor technology. These data can provide valuable insights that can lead to greater efficiencies, productivity, and growth. Moreover, recent regulatory directives, such as those of the United States food and drug administration (FDA), have adopted the concepts of quality by design (QbD) [139] and the strategies of process analytical technology (PAT) [140]. This has resulted in a noticeable shift in the perspective of the industry towards data-intensive process modeling. Since then Design-of-experiments (DoE), a statistical approach, has been used as an excellent tool for engineers and scientists to characterize and optimize bioprocesses. For example, during process design, we can characterize an experiment to discover the effects of variations in process variables on the response or yield. Moreover, the results of characterization will help to identify those control-process variables that can influence and improve the yield. In this thesis, we are exploring the data set of hydrogen (H_2) yield production [141, 142] as a renewable resource for biofuels, which has been obtained using the following DoE methods: Plackett – Burman design [143], the path of steepest ascent, the three variable Box-Behnken design [144], ridge analysis [145], and central composite face-centered design. Interested readers can find an in-depth discussion of these experimental designs and related topics in [146].

The data collected by one or more of these experimental designs then require further analysis in order to evaluate the orthogonality of the design space (i.e., whether relationships between variables are independent) and whether these relationships persist after performing the experiment [147]. Multiple linear regression (MLR) has been used extensively for this purpose. However, this model assumes a linear relationship between input and response variables. Although the inclusion of variables with higher-order polynomials in the model can be advantageous, such a transformation can result in overfitting and reduce the generalizability. Moreover, if the number of observations is relatively few, increasing the number of features by transformation can cause rank deficiency and multicollinearity challenges.

Several excellent data analysis methods exist in the machine-learning field to solve these challenges effectively. However, they have not yet been fully explored in the bioprocess industry. To this aim, in **Publication I**, we explore two state-of-the-art supervised learning methods with a data set, which may open up a new horizon in industrial biotechnology.

5.1.2 Gene expression data analysis

Genes are the basic units that convey the hereditary information in all living organisms. They control the individual characteristics of living organisms by coding *deoxyribonucleic acids* (DNA) for a functional protein or *ribonucleic acids* (RNA). This process is known as *gene expression* [148, 149]. Gene expression includes two steps, transcription and translation. During transcription, information from DNA is synthesized into messenger RNA (mRNA), which serves as a template for protein production. In the next step, information from mRNA is translated into a protein, a key element for essential activities in living organisms. For instance, chemical reactions in the body are performed by enzymes, hormones regulate the functionality in cells while other types of proteins are responsible for transporting materials to organs [148]. Measuring the gene expression can describe the state of a gene of interest. In general, the level of that particular expressed gene is quantified by the amount of mRNA from that gene [150]. The ability to measure gene-expression allows us to compare and identify significantly expressed genes among two or more sample groups. This type of study is known as gene expression profiling and can be used to understand the effect of exposure to varying conditions, such as temperature or the chemical environment. Moreover, we can differentiate between genes that are increased (up-regulated) or decreased (down-regulated) in expression, for example, during heat shock [151, 152].



(a) **(b)**

Figure 5.1: Boxplot representing the variation of microarray data at different temperatures in a logarithmic scale. (a) Unnormalized data, and (b) normalized data.

Microarray, a high-throughput measurement technology, has been widely used for measuring thousands of genes simultaneously. The basic principles and a brief overview of different microarray technologies can be found in [150, 153]. This technology has had a significant impact on biomolecular research, such as in explaining biochemical pathways, the diagnosis of a disease and its stages, and giving guidance for drug design and discovery [150]. For instance, the gene expression profiling of a food-borne pathogen, *Vibrio (V.) Parahaemolyticus* can identify differentially expressed genes in response to changes in temperature. These pathogens are often exposed to different environmental signals (such as temperature, pH, nutrient availability, oxygen, and iron levels) during an infection cycle and they can alter the gene expression for survival and growth [154–156]. To this end, in this thesis, we have studied the behavior of *(V.) Parahaemolyticus* RIMD 2210633 under different thermal conditions (see **Publication II**). However, the analysis and handling of

microarray data is becoming a major challenge due to the high-dimensional characteristic of the data. Additionally, the accurate correlation between mRNA levels and the observed protein levels are sensitive to noise due to the limitations of the measurement technologies and the complex nature of the gene expression process [157]. Therefore, fast, robust, accurate data-processing pipelines are required for further analysis and correct biological interpretation.

In gene expression analysis, the data-processing pipeline, as shown in Figure 2.1, is quite sophisticated and requires careful examination. The challenges with the raw data include missing values, nonbiological biases and variability. In order to remove these biases and make the data comparable, the first step is the data preprocessing, which involves background correction, filtering, and normalization. Figure 5.1 illustrates an example of the gene expression data used in **Publication II** before and after the normalization process. The next step is to seek the differentially expressed gene from the preprocessed data using unsupervised methods, such as clustering as discussed in Section 3.2. The later, more challenging, part involves linking the results to functional annotations and biological processes. One excellent source for the data workflow in gene expression analysis can be found in [158].

5.1.3 Flow cytometry data

Another application area of machine learning is flow cytometry data analysis, which is used for rapid qualitative and quantitative analysis of individual cells [12]. Flow cytometry is widely used as a biomedical research tool in immunology and cancer biology to distinguish different cell types in mixed populations. It is also used in clinical laboratories as a diagnostic tool for diseases, such as leukemia, lymphoma, and abnormal proliferation of lymphocytes [159–161]. A flow cytometer measures the physical and chemical properties of individual cells. Thus, this tool is used to identify particular cells that can be used to distinguish, for example, between Acute Myeloid Leukemia (AML)-positive patients and healthy donors. In flow cytometry analysis, cells are labeled with fluorochrome-conjugated antigens that bind to cell surface and targeted proteins. The attributes or features that are measured by a flow cytometer are scatter and fluorescent parameters of the cells. The scatter parameters are typically forward scatter (FSC) and side scatter (SSC), which correspond to cell size and cell granularity, respectively. The fluorescent parameters are the expression of different antigens on the cell surface. In the measurement process, FSC and SSC are always measured, as these control parameters distill the cells with corresponding features by a process called *gating*, for subsequent analysis. The fluorescent markers are part of a DoE process and the number of measurements can vary for different experiments. For example, the presence of the CD45 biomarker in a human blood cell is expressed when combined with the CD45-ECD antibody and this marker is important in identifying AML cells from normal ones [162].

Through recent developments in laser technology, the advanced flow cytometer measures up to 18 markers on an individual cell simultaneously. However, in most clinical settings, the modern flow cytometer still only measures $5-7$ biomarkers at once [163]. For instance, each donor's sample in the data studied in **Publication III** was divided into 8 tubes (aliquots) as shown in Figure 5.2. Each tube is stained with 5 different combinations of fluorochrome-conjugated antigens.

Figure 5.2 presents a simple flow cytometer, where the fluorochrome-labeled cells are passed through a laser beam and photo detectors detect the scatter and fluorescent parameters. These parameters are digitized and stored in a computer file system. Each

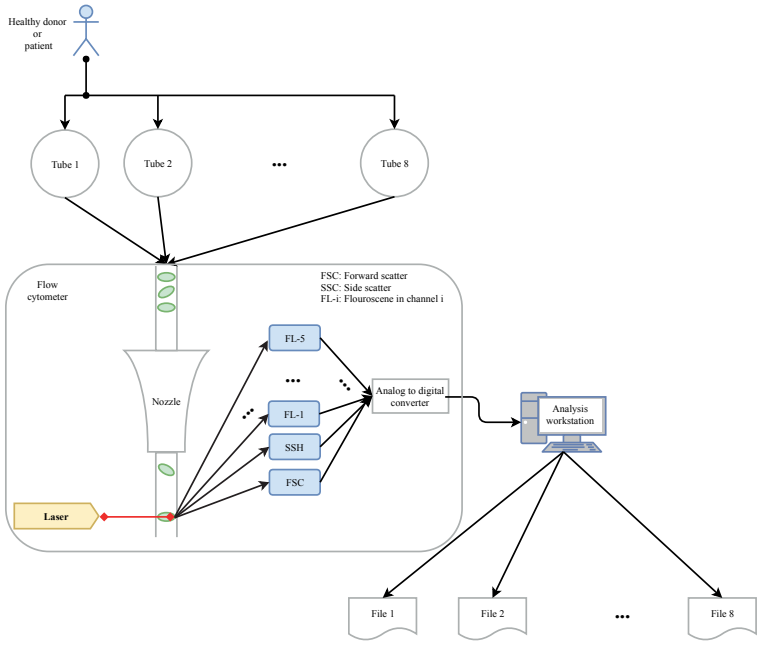


Figure 5.2: A simplified version of flow cytometry analysis. Image adapted from Figure 1 in [160].

file system stores the measured scatter (FSC and SSC) parameters. The other parameters stored in a file correspond to different antibodies targeting different antigens in a particular tube (See Table 1 in **Publication III** as an example). The number of cells per tube can vary. For instance, the data that we used in our analysis contained $28,853$ cells in Tube 1 and $28,610$ cells in Tube 2, which were collected from the blood sample of a single donor. A flow cytometer can process from $10,000$ to $1,000,000$ cells in a single session [12, 164]. Traditional analysis of such large quantities of measurement data involves manual gating through one- or two- dimensional plots at a time. This task is subjective, time consuming, and thus ineffective for high-dimensional data [165].

For the aforementioned reasons, there is a growing need to develop a reliable automated approach to flow cytometric analysis. Thus, in 2010, a project called Flow cytometry: Critical assessment of population identification methods (FlowCAP) was initiated to promote the development of computational methods to identify cell populations of interest in flow cytometry data. Later, in 2013, FlowCAP organized a molecular classification challenge of flow cytometry data in conjunction with another consortium, Dialogue for Reverse Engineering Assessment and Methods (DREAM) [166]. In the challenge, several sophisticated supervised classification algorithms were presented that achieved 100% accuracy on the given data set. In this thesis, we concentrate on a simple feature extraction technique to obtain obvious features while still retaining good accuracy in a $p \gg n$ setting.

5.1.4 Bacterial cellulose synthesis

Cellulose is the most abundant organic polymer on earth and it is synthesized by both plants and bacteria [167]. At a molecular level, it is a polysaccharide (a long chain of sugar or glucose molecules bonded together) composed of linear chains of $\beta - 1,4$ D-glucose or glucan units. These linear chains are arranged in parallel and crystallized through hydrogen bonding to form fiber-like nanostructure strands, known as *cellulose micro-fibrils*, with high tensile strength [168]. The aggregation of micro-fibrils can form larger structures, which differ greatly among organisms. The structure is measured by the number of glucan units, known as the *degree of polymerization* (DP). The DP of cellulose micro-fibrils in higher plants, for instance, ranges from 100 to 15,000 units, while in bacteria, the DP of cellulose micro-fibrils is 100 to 10,000 units long [169–171]. Natural cellulose is found in two crystalline forms, which corresponds to the position of the hydrogen bonds within and between the micro-fibrils. Cellulose I is the most abundant form that exists in nature, where the glucan chains are oriented in parallel. In cellulose II, the orientation in the glucan chains is antiparallel. This is rare in nature and found only in algae and bacteria. Cellulose II is thermodynamically the most stable allomorph and the conversion from cellulose I to cellulose II is irreversible. Other types of crystalline structures are cellulose III and IV, which can be produced from cellulose I and cellulose II by chemical treatments [167, 172–174]. The crystalline structure of cellulose I is a mixture of two different suballomorphs: cellulose I_α (triclinic) and cellulose I_β (monoclinic) [175]. The pure form of cellulose I_β is rarely synthesized naturally and it is found mostly in higher-level plants. On the other hand, cellulose I_α is dominant in bacteria. Being an insoluble biopolymer, cellulose has been used as a renewable raw material and approximately 150 billion tons of cellulose is synthesized annually [171, 176]. The polymeric structure of cellulose explains its properties, for instance, hydrophilicity, biodegradability, broad chemical-modifying capacity and its formation of versatile semicrystalline fiber morphologies. Thus, cellulose can be used as a platform for developing new biomaterials and products that meet the requirements of environmental legislation.

In recent years, bacterial cellulose has become a promising candidate for various applications due to its high stiffness, low weight, purity, and enormous water-holding capacity [177, 178]. Bacteria produce the extracellular materials in order to protect themselves from the negative effects of ultraviolet radiation, harsh chemical environments, and to provide access to oxygen [167, 170, 179]. Although plants are the major resources for the industrial production and processing of cellulose, extra processing is required to remove impurities and contamination [167, 178]. Interested readers are referred to [171, 180–188] to get an overview of the excellent fundamental and useful properties that provide bacterial cellulose with a variety of promising applications. These include artificial skin, vascular tissue engineering and wound dressing materials in the biomedical field, reinforcement agents in high-quality paper, sewage purification, cosmetics, paint additives, pharmaceuticals, and diaphragms in electro-acoustic transducers [178–180].

Thus, such high-value bacterial cellulose has become the subject of continuous research from pilot-scale to commercial production. For bacterial cellulose synthesis, a cyclic nucleotide, known as cyclic-di-guanosine monophosphate (c-di-GMP) serves as an allosteric activator [189]. This activator is regulated by several proteins. Among them, the GGDEF, EAL, HD-GYP, PilZ, and MshEN proteins are involved in metabolic activities, such as biofilm formation and motility. For example, the GGDEF and EAL domain regulates the synthesis and degradation of c-di-GMP. PilZ and MshEN act as receptors while

HD-GYP is involved in hydrolyzing activities [190]. Although these proteins are encoded in the bacterial genomes from diverse branches of the phylogenetic tree of bacteria, not all bacteria participate in cellulose synthesis. Moreover, studying these effectors and targets of c-di-GMP across diverse bacteria is still obscure. In this thesis, we study the phylogenetic distribution of the c-di-GMP signaling domains across diverse bacteria and use machine learning to identify the supporting pathways associated with those proteins.

5.1.5 Magnetoencephalography data

Information processing in the human brain, or more specifically, the way information is perceived by the human brain is always an interesting field of study. This is due to modern functional neuroimaging methods, which use radioactive materials or magnetic sensors to measure the brain activity. These include *electroencephalograms* (EEG), *positron emission tomography* (PET), *functional magnetic resonance imaging* (fMRI), and *magnetoencephalography* (MEG) [191 – 195]. The use of machine learning in this emerging field has added several data-intensive tasks, such as dimensionality reduction, classification, pattern discrimination and pattern localization to answer many brain-imaging questions of interest [196]. In this thesis, we explore MEG measurement data from the model assessment point of view.

In MEG, the brain activities are mapped by recording the magnetic fields generated by electric currents occurring in a human brain. In our experiment, we use the measurement data available in the *Mind reading from MEG* challenge, which was organized together with the International Conference on Artificial Neural Networks (ICANN) 2011 conference[1]. The primary task of the challenge was a 5-class classification problem, where the aim is to infer the type of movies being shown to a subject based on the MEG measurements. The organizer also considered a binary classification problem: separating the videos which have a plot from those without one. The measurements were recorded in two different sessions. The approach of the winning team [197] extracted 408 features from the original data that we have considered for our experiment.

5.2 Summary of the research efforts

This section summarizes the applicability of machine-learning approaches in biological data analysis, which solve the associated challenges mentioned in the previous section. Each sub-section aims to achieve the objectives listed in Chapter 1.

5.2.1 Regularization approaches in biological case studies

Multiple linear regression is often used as a tool to fit a linear model in bioprocess data analysis. However, the limited number of observations compared to the number of features (i.e., $p \gg n$) makes the task an ill-posed problem and MLR may not be able to provide a model with accurate predictive capabilities. One interesting experiment in the bioprocess data analysis is to study the interaction between the features by transforming the feature space to higher order polynomials, for example, a quadratic polynomial. To this end, we set up an experiment with the bioprocess data in **Publication I**. The first experiment is performed with the original data set. Figure 5.3 shows the results for fitting the model with MLR and lasso. In the second experiment, we use the data that had

[1] http://www.cis.hut.fi/icann2011/mindreading.php

been transformed using a second-order polynomial function. The result is illustrated in Figure 5.4.

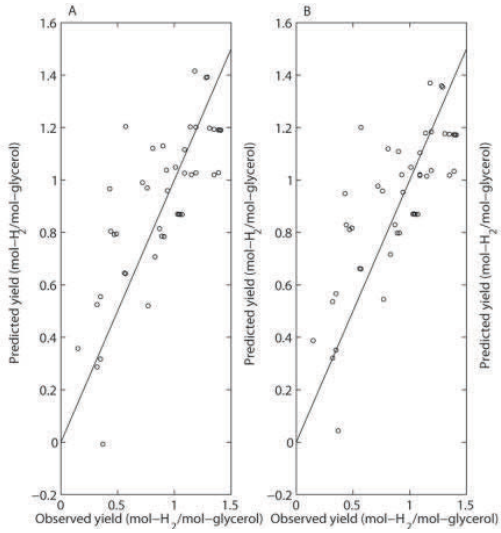


Figure 5.3: Comparison of prediction performances for the original data set. A) multiple linear regression, and B) lasso. The straight line illustrates the position of the perfect prediction.

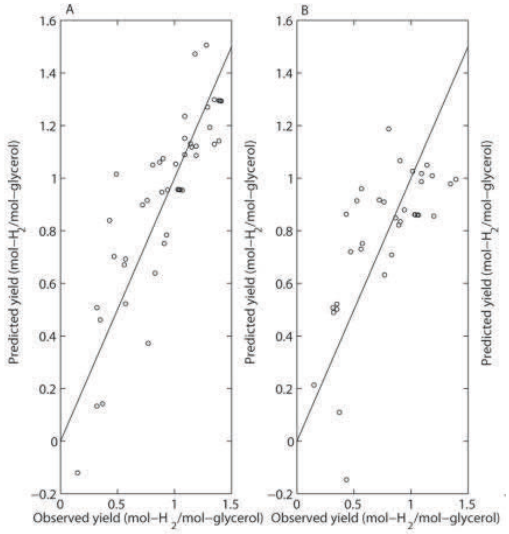


Figure 5.4: Comparison of prediction performances for the quadratic data set. A) multiple linear regression, and B) lasso. The straight line illustrates the position of the perfect prediction.

The correlations (ρ) between the true response and the predicted response can be measured using the correlation coefficient. These are summarized in Table 5.1. This table shows a comparison of the different methods which reveals that the inclusion of more features has only an insignificant effect on the regularized approach, even though there is a slight improvement.

Table 5.1: The correlation coefficient ρ for different predictive models.

Models	ρ
Linear MLR	0.65
Quadratic MLR	0.012
Linear Lasso	0.60
Quadratic Lasso	**0.69**

We also use the regularization approach to fit a model in **Publication IV** with bacterial cellulose synthesis data. Figure 5.5 illustrates the prediction performance of the lasso and random forest models for gene distribution in five different domains of cellulose synthesis. A novel finding in this experiment is that the variance is quite high for the EAL and MshEN domains. Moreover, lower correlation (R^2) values are also reported for these two domains. This result highlights how little is known about these two domains. In fact, GGDEF is the most studied of all the c-di-GMP signaling proteins, whereas MshEN is a recently discovered domain. In this experiment, superior results are seen for the random forest method. However, the sparsity properties of the regularization approaches actually provide a better interpretability of the learning model.

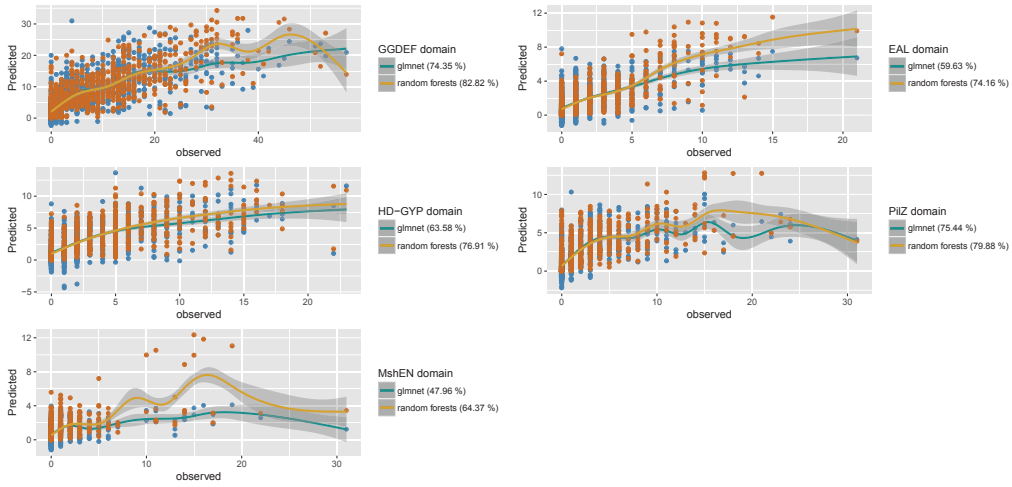


Figure 5.5: Prediction of distribution of genes encoding c-di-GMP signaling domains.

5.2.2 Unsupervised learning in high-throughput data analysis

The aim of the clustering is to seek a group of genes that are expressed differentially. In **Publication II**, hierarchical clustering has been used to determine the similarity of the replicated samples at different temperatures. Figure 5.6 illustrates an example of a tree-like graph, called a dendrogram, which is obtained from the hierarchical clustering approach.

The interpretation of the dendrogram shown in Figure 5.6 is straightforward. The replicates at the temperature of 42°C form a clear cluster. Replicates at temperatures of 20°C and 37°C also form clusters. The replicates at four other temperatures are correlated to each other. The dendrogram also identifies the outlier-replicated samples,

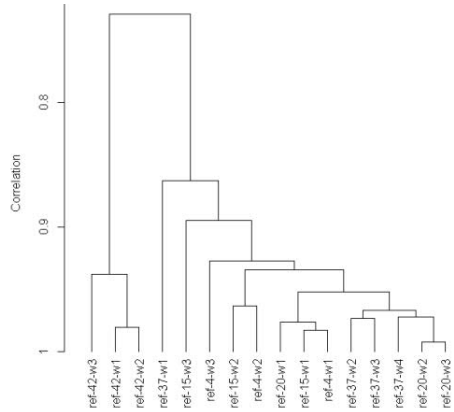


Figure 5.6: The similarities of replicate samples are represented by the dendrogram. The x-axis represents the sample label with different temperatures and the replicate number at the corresponding temperature. The y-axis represents the correlations among the samples.

such as ref-37-w1, and ref-20-w1. These two samples are clustered with the replicates from different temperatures. Therefore, we can interpret that when compared with the 37°C samples, there will be several differentially expressed genes in the 42°C samples, slightly fewer in the 4°C and 15°C samples, and even fewer in the 20°C samples. Moreover, differential expression analysis has been done using linear models provided by `limma` package [198]. Table 5.2 lists the number of differentially expressed genes and the probe sequence for each temperature to the reference temperature (37°C).

Table 5.2: Comparisons in differential expression analysis, along with the number of differentially expressed genes and probe sequences.

Comparison	Number of differentially expressed genes	Number of differentially expressed probe sequence
4°C and 37°C	194	324
15°C and 37°C	660	1309
20°C and 37°C	19	33
42°C and 37°C	641	1197

We demonstrate an example of the heatmap of differentially expressed genes at the temperatures of 37°C and 20°C. Additionally, a hierarchical clustering of the genes and the probe sequences is also presented. In Figure 5.7, down-regulated genes are expressed in green while up-regulated genes are expressed in red.

5.2.3 Feature extraction and selection

The challenges associated with the flow cytometry data are: i) the number of cells in each tube is arbitrary, ii) a different combination of biomarkers is measured from each tube, and iii) the number of data points is large compared to the number of observations. Therefore, we need a good feature-extraction approach to construct a good model. These issues are addressed in **Publication III**. Different variations of feature extraction-techniques along with ours are listed in Table 5.3.

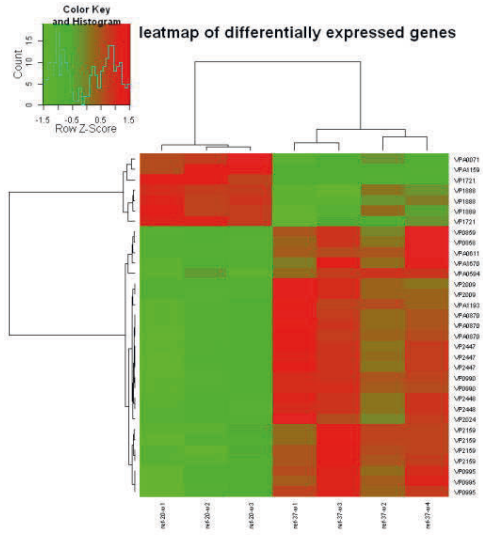


Figure 5.7: Heatmap presents differentially-expressed genes by comparing samples from the temperatures of 37°C and 20°C.

Understanding a biological process requires not only a predictive model but also a set of salient features, which describes the relationship with the response. Thus, the feature-selection process also plays a significant role in the model construction. In this thesis, we use ℓ_1 regularization methods for fitting a model, which also serves as an automatic feature-selection technique. **Publications I** and **IV** work towards this. The significant features and their associated coefficient values are listed in Table 1 and Table S1 of **Publication I**. The feature-selection approaches used in **Publication IV** assist in understanding the data and emphasize the pathways associated with c-di-GMP binding proteins in bacterial cellulose production. This is illustrated in Figure 5.8. In the experiment, lasso fails to associate any pathway to the MshEN domain (gray). On the other hand, the relevance of the rest of the pathways associated with the domains agree with our hypotheses (see Table 1 in **Publication IV**).

5.2.4 The goodness of feature-selection approaches

The goodness of a feature-selection method depends on the stability and the robustness of the feature-selection process. These issues are addressed in **Publication III**, where we study the performance analysis and the robustness of supervised machine-learning approaches relative to the number of features in a $p \gg n$ setting. Here, we observe the behavior of feature-selection criteria, which is influenced by the characteristics of the data, such as the dimensions and the number of observations [202]. We illustrate in Figure 5.9 the phenomenon with a data set of $p \gg n$. The aim of the feature selection is to gain consistency in the number of features selected irrespective of the quantity of training data. However, real-world applications including our own, show that the variations in selected features differ as the number of training data varies. Moreover, this variation is influenced by the performance measurement metrics used for model selection. For instance, in Figure 5.9 (a), 10-fold cross validation has higher variability than the BEE estimator with proper prior. Additionally, we study the robustness of the feature-selection

Table 5.3: Studies based on feature-extraction strategies for the DREAM AML challenge data set. The AUROC is measured for a single train/test split.

	Accuracy	Size of feature vector	Brief description
Biehl et al. [199]	1.00	186	Extraction of features with moments, median and interquartile and learning vector quantization is used for prediction
Vilar et al. [200]	1.00	31	Extraction of features with entropies and histogram based classifier is used for prediction
Manninen et al. [201]	1.00	(# of events) x 84	Expand features to higher dimension and then mapping to 1-D using linear discriminant analysis; logistic regression is used for prediction
Publication III	0.9989	49	Extraction of feature vector from means of measurements and applying regularized logistic regression for prediction
Publication III	0.9992	98	Extraction of feature vector from means and standard deviation of measurements and applying regularized logistic regression for prediction

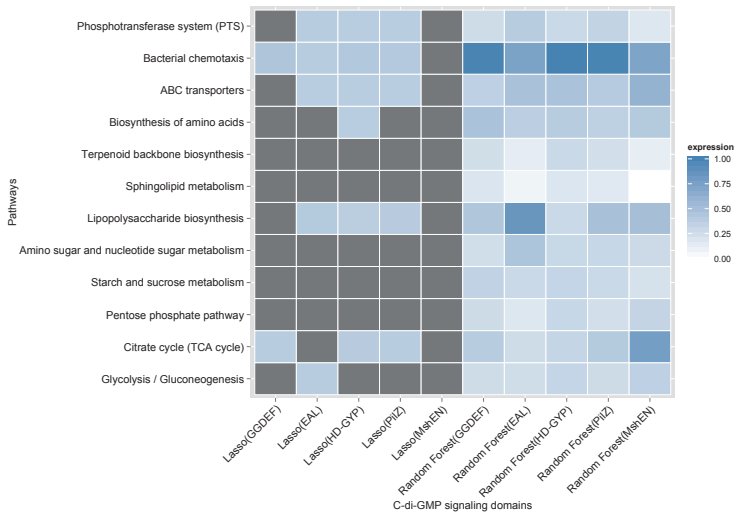
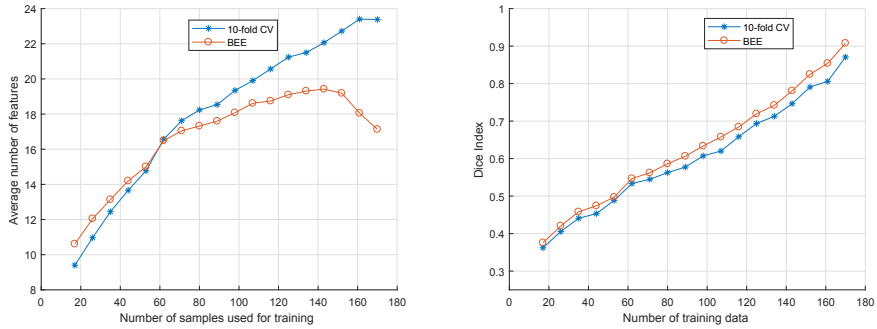


Figure 5.8: Heatmap representation of significant pathways in c-di-GMP signaling domains. The gray color shows that there is no association between the protein domains and the pathways.

process in **Publication III** using the Sørensen-Dice coefficient metric. The probability of choosing a similar subset of features between two different iterations is illustrated in Figure 5.9 (b). The experimental result shows that BEE has a higher degree of stability than that of CV.

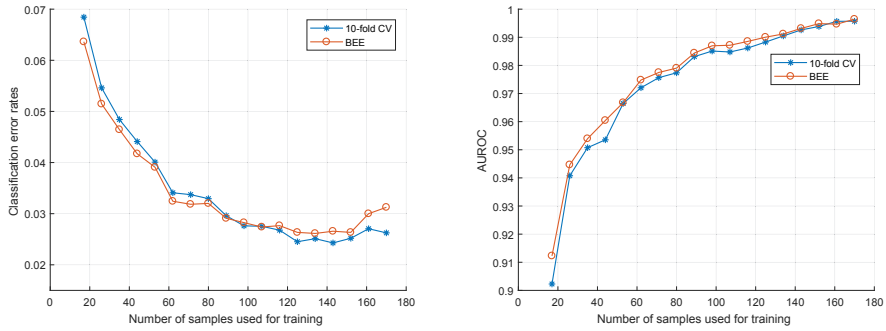


(a) **(b)**

Figure 5.9: (a) Comparison of the number of selected features for 10-fold CV and BEE with proper prior with $p = 98$. (b) Stability measures for 10-fold CV and BEE with proper prior with $p = 49$.

5.2.5 Accuracy assessment

The accuracy assessment is used to quantify the performance of a model, to evaluate the generalization ability of that model, and for the selection of the model. Both cross-validation and BEE are iterated through a set of hyperparameter values of the model. Our results show that BEE is a more accurate error estimator than cross-validation. In **Publication III**, we study the classification task in flow-cytometry data and we consider two performance measurement metrics: classification error rate and AUROC. The result is illustrated in Figure 5.10.



(a) **(b)**

Figure 5.10: Comparison of performance measurement metrics for 10-fold CV and BEE with proper prior with $p = 49$. (a) Classification error rates vs. the number of training samples. (b) AUROC vs. the number of training samples.

In **Publication III**, we used regularized logistic regression for classification. However, we extended the study of the accuracy estimation for a generalized linear model in **Publication V**. Here, we consider the area under the receiver operating characteristic curve (AUROC) as an accuracy measurement metric. The cross-validation AUROC (CVAUROC) is estimated by first splitting the training data into K-folds and using each fold as a validation set. The resulting K-fold estimates are then averaged to produce a final

CVAUROC estimate. On the other hand, an empirical Bayesian AUROC (EBAUROC) uses a detection threshold which slides in the ROC space. The true positive rate (TPR) and false positive rate (FPR) are then estimated using BEE. In order to speed up the computational time by avoiding iterative steps, we proposed a closed-form expression for BEE (CBAUROC) in **Publication V** (see Section 4.4.2). In our experiment, we evaluate the performance of a model in a binary classification task using simulated data. The idea is to study the behavior of the different estimators when the number of training observations varies. For this purpose, the samples for two classes are randomly drawn from p-dimensional normal Gaussian distribution with a common covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ and means $\boldsymbol{\mu}_1 \in \mathbb{R}^p$ and $\boldsymbol{\mu}_2 \in \mathbb{R}^p$. Figure 5.11 illustrates examples of the behavior of different error estimators with dimensions $p = 4, 10, 100$ and $\Sigma = I, \boldsymbol{\mu}_1 = \mathbf{0}$, and $\boldsymbol{\mu}_2 = \mathbf{1}$.

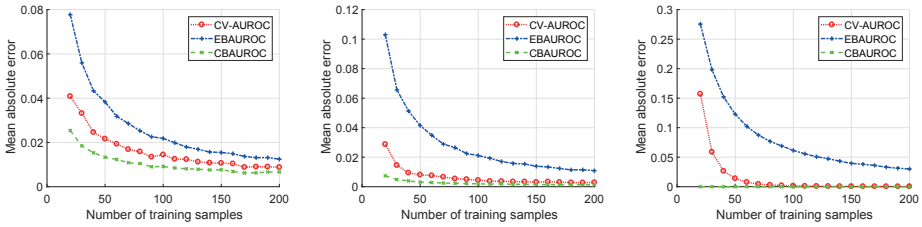


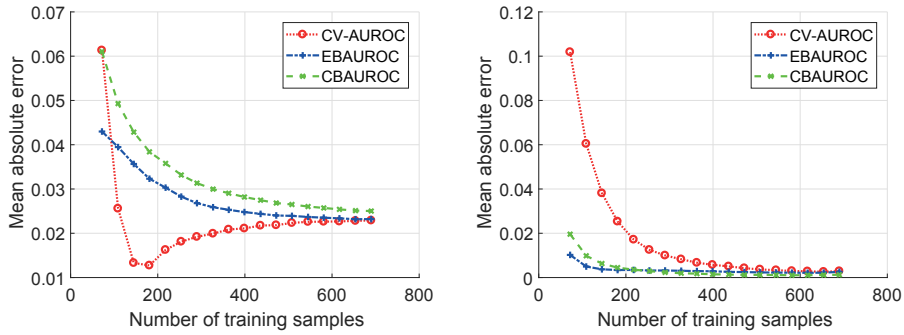
Figure 5.11: Accuracy assessment using multivariate Gaussian data. The mean absolute error (MAE) of three AUROC estimators to the true AUROC with dimensionality (a) $p = 4$, (b) $p = 10$, and (c) $p = 100$. The error estimators are 5-fold cross-validation AUROC (CVAUROC) (red curve), empirical Bayesian AUROC (EBAUROC) (blue curve) and proposed closed-form Bayesian AUROC (CBAUROC) (green curve).

In real-world applications, the data are generally not evenly distributed among the classes. Moreover, the error estimation of a model becomes difficult with the discrepancy between the training and test set, and the non-Gaussian distributions of the underlying data-generating process. These findings are also reported in **Publication V**. As a case study, Figure 5.12 illustrates the phenomenon using MEG data. The results show that the performance of CVAUROC is more pronounced than the other alternatives.

On the other hand, the performance of our proposed method is weaker than the other two estimators. Moreover, we can see an obscure shape for the CVAUROC estimator in Figure 5.12 (a). This result suggests that the CVAUROC would be the most accurate with training sample size of about 150. However, the accuracy degrades as more samples are added. This clearly indicates that there is an effect of the discrepancy between the train and test subsets. In fact, the two sets are measured from the same subjects however, in different days. Therefore, the characteristics of the measurements could vary during different days depending on the mental state of the subject. Moreover, the organizers also added a small amount of test data into the training data. Since the CVAUROC estimator is depended entirely on the data, this estimator can exploit these samples and hence, the model is overfitted.

We also explore the Bayesian inference in Figure 5.12 (a) and Figure 5.12 (b). The results show that the mean absolute error is lower for the empirical estimator than that of the CBAUROC. This is because our proposed method is “fully Bayesian” and this estimator considers only the statistical properties (means and covariances) of the data. On the other hand, the EBAUROC uses the observations during the iteration through the thresholds,

and hence it has lower error rates. We can interpret EBAUROC is "less Bayesian". Furthermore, the variance of CBAUROC is the lowest among the three estimators (See Figure 1(d)-(f) and Figure 2 of **Publication V**).



(a) **(b)**

Figure 5.12: Accuracy assessment using ICANN 2011 MEG binary data set. The mean absolute error (MAE) of three AUROC estimators to the test AUROC (a) using the predefined train/test split and (b) using hold-out test data within the training set. The error estimators are 5-fold cross-validation AUROC (CVAUROC) (red curve), empirical Bayesian AUROC (EBAUROC) (blue curve) and proposed closed-form Bayesian AUROC (CBAUROC) (green curve).

5.3 Discussion

This thesis focuses on finding an optimal statistical model to answer a real-life biological question and evaluating the accuracy of that model. Finding a model and evaluating it are straightforward problems in machine learning, yet high-dimensionality in the feature space and a limited number of observations complicate the tasks. The situation is further complicated by, for example, the imbalanced distribution of classes in the classification task, the discrepancy in the training-test split, and samples drawn from non-Gaussian distributions.

Regularization approaches are useful in solving high-dimensional problems. Moreover, these approaches can lessen the danger of overfitting and improve the generalization ability of a model. This motivated us to design the experiments and provide statistical models for bioprocess data analysis and bacterial cellulose synthesis. The results are reported in **Publications I** and **IV**.

However, different machine-learning methods are required to solve problems associated with unlabeled data, such as gene expression data analysis. Careful examination and data preprocessing are necessary for further analysis. Then, an unsupervised method, such as clustering is used to seek similarly expressed genes from the preprocessed data. The next step involved in gene expression analysis is to fit a linear model. Finally, differentially expressed genes are linked with their functional annotation and biological processes using an annotation tool. **Publication II** addresses all these steps.

Regularization approaches, such as ℓ_1 regularization, are an attractive choice for feature selection. The selection of features helps us to understand which features are salient for answering a biological problem of interest. These are observed in **Publications I**, **III**,

and **IV**. In the feature-extraction pipeline, we use the flow cytometry data. These findings are reported in **Publication III**. We observe that simple feature-extraction techniques, such as a statistical summarization of the data, can also achieve similar accuracy as other competing approaches. In fact, we can discard 50% of the features that have a very insignificant contribution to the response.

Cross-validation is a traditional approach for model hyperparameter selection and for avoiding overfitting, and this is studied in **Publications I** and **IV**. However, the instability in feature selection is observed in cross-validation due to the randomness characteristic in the training-test split operation. We can avoid randomness by using leave-one-out cross-validation, yet this again introduces an overfitting challenge. This motivated us to study the stability and the robustness of the feature-selection approaches, which are reported in **Publication III**. We have shown that BEE is more stable than cross-validation in feature selection.

Moreover, in **Publication I**, we measure the correlation between the true response and the estimated response using LOOCV. The selection of hyperparameter values is done with both manual selection and automatic selection. We observe a large gap between the manual selection of the hyperparameter values and the proper cross-validation. This implies that there exists an inaccuracy in the error estimate by cross-validation when the number of observations is limited. We also notice the randomness in the K-fold CV as illustrated in Figure 1 of **Publication III**. Furthermore, we had to retrain the model at each fold and an additional retraining for the whole data set was also required.

These aforementioned findings raise the necessity of seeking an alternative approach that is computationally faster, and more stable and robust in small sample settings. As a consequence, we introduce an alternative accuracy metric in **Publication V**. For the sake of simplicity, we consider a binary classification task with the data drawn from a Gaussian distribution with a common covariance structure. This sets a limit to our proposed method and the sensitivity towards this Gaussian homoscedastic distribution is reported in Figure 3 of **Publication V**. We also consider the effect of class imbalance, and the stability of our method is more pronounced than the other alternatives (see Figure 4 of **Publication V**).

A gap in the training and test set is also an interesting phenomenon to study. In this regard, we studied the MEG data in **Publication V**. Figure 5.12 illustrates the accuracy assessment of the error estimators. Here, we see our proposed method is unsuccessful in obtaining the optimal result. The performance of the CVAUROC estimator is more pronounced. In fact, we observe an obscure curve in Figure 5.12 (a). Even though the data was collected from the same subject, the training and test sets were measured on two different days. Moreover, a small portion of the training set contains data from different days. As a consequence, we notice the discrepancy in the training-test split which affects the results. These concerns will form the basis for future study.

The variation in the error estimates has also been studied in **Publication V**. An error estimator with a low variability with respect to the true error rate is a desirable quality. Our proposed method is a good choice for a small number of samples in this context, since it has lower variability than the other methods (see Figure 1 (d)-(f) of **Publication V** as an example).

The assessment of the error estimators in this thesis is limited to 5-fold cross-validation, and Bayesian estimators. The scope of the bias-variance decomposition has been addressed only for these estimators. Different variations of K-fold cross validation and other

estimators along with the bias-variance decomposition will be a part of possible future research direction.

6 Summary of publications

This section summarizes the publications presented in this thesis and the author's contributions to these publications.

6.1 Overview of publications

Publication I In this article, we study the feasibility of statistical modeling in bioprocess data in order to understand the relationship between the control parameters and the production yield. The study also includes identifying the primary control parameters and determining a useful control direction for maximizing the yield. As an example, we use data from a culture media optimization study for microbial hydrogen production. The initial data set is obtained using design-of-experiments methods. The data set is then examined using regularized regression (lasso) and random forest and the performance of these machine-learning methods is benchmarked against multiple linear regression. The results show that all three methods are capable of producing effective models when the parameters are linearly correlated to each other in modeling. However, multiple linear regression fails to comprehend the inclusion of higher-order correlation between the parameters in modeling, while the modeling is successful with lasso and random forest. The results show that both lasso and random forest were able to produce feasible models, and the latter was efficient in capturing any non-linearity in the data. In this kind of data mining task with bioprocess data, both the studied methods outperform the traditional multiple linear regression approach.

Publication II In this article, we study the gene expression profiling of a food-borne pathogen, Vibrio Parahaemolyticus RIMD 2210633, in order to understand the behavior of this pathogen in cold and hot temperatures. The experiment was performed via microarray at five different temperatures (4°C, 15°C, 20°C, 37°C, and 42°C). For the gene-expression analysis, 37°C is used as a reference temperature. The gene expression values are benchmarked against RT-qPCR experiments by measuring the coefficient of determination on a logarithmic scale. Moreover, the differentially-expressed genes at each temperature are visualized using a volcano plot, which also aids in assessing the hybridization qualities and the comparability of the data. Finally, an annotation tool, the Database for Annotation, Visualization and Integrated Discovery (DAVID), has been used for functional annotation to quantify the genes sharing similar biological processes. The results show that the largest number of significantly expressed genes are observed at 15°C and 42°C with 13.3% and 13 %, respectively. Moreover, the genes of many functional categories were highly regulated even at lower temperatures. The results also confirm that the

genes associated with pathogenicity are unaffected by the changes in temperature. This study demonstrates that the analysis of gene-expression of such pathogens can reveal the response of metabolistic alteration in conditions like storage and transport.

Publication III In this article, we study the classification problem in flow cytometry data analysis from the feature-selection point of view. The existing machine-learning algorithms can effectively reach 100% classification accuracy given enough training data without any prior knowledge of the underlying biological process. The flow cytometry data set used in this experiment includes large quantities of partially redundant measurements. Thus, an automated feature-selection process is required in order to promote generalization of the classification model and to avoid overfitting. Earlier studies include sophisticated machine-learning algorithms with complex feature-extraction pipelines. We focus on using simple feature-extraction techniques in the classification tasks and search for feature-selection approaches that are reliable, even with extremely small sample sizes. Our results show that 50% of the features can be discarded without compromising the prediction accuracy. We also study the model-selection problem in logistic regression classifiers with a recently-proposed Bayesian error estimator. We observe the variation in selecting features with respect to the number of training data. The variation is influenced by the performance measurement metrics used in model selection. We show that the Bayesian error estimator has higher stability in selecting features than the traditional cross-validation approach.

Publication IV In this article, the aim is to find the significant pathways for five essential c-di-GMP encoding proteins in bacterial genomes. We study the feasibility of two machine-leaning approaches for modeling and selecting the critical pathways required in bacterial cellulose production. Both the regularized linear model (lasso) and tree-based model (random forest) show that bacterial chemotaxis is the most essential pathway for c-di-GMP encoding proteins. However, strong regularization of the lasso model failed to associate any pathway with the MshE domain, which is a recently discovered c-di-GMP binding protein involved in biofilm formation in bacteria. Results from the analysis may help to understand and emphasize the supporting pathways involved in bacterial cellulose production. These findings demonstrate the need for a chassis to restrict the behavior or functionality by deactivating the selective pathways in cellulose production.

Publication V We propose a novel classifier accuracy metric: the Area Under the Bayesian Receiver Operating Characteristic Curve (CBAUROC) based on the recently proposed Bayesian minimum mean square error estimator. This new metric can assess the quality of a classifier using only the training data set without an upward bias and without the need for computationally-expensive cross-validation. We derive a closed-form solution of the proposed accuracy metric for a linear two-class classifier under the Gaussianity assumption, and study the accuracy of the proposed estimator using both simulated and real-world data. These experiments confirmed that the closed-form CBAUROC is both faster and more accurate than conventional AUROC estimators.

6.2 Author's contribution

This research work was carried out in collaboration at the laboratory of signal processing and the laboratory of chemistry and bioengineering in Tampere University. The work was supervised by Assoc. Prof. Heikki Huttunen, Prof. Olli Yli-Harja, Prof. Matti Karp, Assist. Prof. Ville Santala, and DSc. Tommi Aho. All the publications are the outcome of collaborative supervision. The author is the main contributor to **Publications I, III, IV, and V** and also contributed to **Publication II**. A brief description of the contribution of the author to each publication is stated below:

Publication I The author of this thesis was responsible for interpretation of the data, the design and analysis of the models, and made a substantial contribution to writing the manuscript in joint collaboration with Muhammad Farhan. Rahul Mangayil was responsible for the acquisition of the data. Tommi Aho and Heikki Huttunen contributed to the design of the study, and in writing and revising the manuscript.

Publication II The author of this thesis assisted in the data analysis and in revising the manuscript. The data analysis part includes preprocessing, finding similar groups, and postprocessing with annotation tools. These steps are significant for making a critical analysis of the study for publication. Sara Urmersbach conducted and performed the RT-PCR, qRT-PCR and microarray experiments. Thomas Alter and Stephan Huehn participated in the study design, data analysis, manuscript drafting and editing. Tommi Aho and Reija Autio assisted with data analysis and manuscript revisions.

Publication III The author of this thesis implemented the code, designed the experiments and was responsible for writing and revising the manuscript. Pekka Ruusuvuori and Heikki Huttunen contributed to the implementation of the code, designing the experiments, writing, revising, and reviewing the manuscript. Leena Latonen was responsible for the acquisition of the second data set and contributed to writing and revising the manuscript.

Publication IV The author of this thesis was responsible for the design of the study, preparation of the data, the implementation, the analysis of the results and writing, and revising the manuscript. Rahul Mangayil was also responsible for the analysis of the results and writing the manuscript. Tommi Aho, Olli Yli-Harja and Matti Karp contributed to manuscript revision.

Publication V The author of this thesis was responsible for designing the concept, developing and implementing the theory, verifying the theoretical formulation, analyzing the results, and writing and revising the manuscript. Heikki Huttunen contributed to the design of the concept, the formulation and verification of the theory, and writing and revising the manuscript. Jari Niemi assisted with formulating the theory while Jussi Tohka contributed to revising the manuscript.

7 Conclusions

This thesis has provided a comprehensive analysis of the applicability of machine-learning approaches in biological data. Our results indicate that the integration of regularization in simple and straightforward machine-learning approaches is highly relevant for heterogeneous applications in biology. In general, high-dimensional biological data pose challenges since the data include a large number of features but a relatively small number of samples, yielding an ill-posed problem for data analysis. The purpose of regularization is to add additional information that can transform an ill-posed problem into a well-posed problem. We can infer following conclusions based on the publications in this study:

- The regularization approach was very effective in bioprocess data to discover the relationship of process variables, which were the concentration of nutrients (such as carbon, oxygen) to the response, i.e., the fermentation culture. Regularized linear regression was shown to outperform conventional methods for high-dimensional data with redundant features. The regularized regression was also performed as an efficient feature selector to identify primary control process variables for both influencing and improving the response. However, the superiority of ensemble methods like random forest was more pronounced with the nonlinear relationship between variables and the response. Our results indicated that, for the given dataset, the correlation between the true and predicted responses was 31.88% higher for random forest with default parameters than for the regularized regression. Of course, these findings do not mean that the random forest would perform better on all datasets without tuning the parameters, and nor can dismiss the advantages of the explanatory capabilities of regularized regression.

- We presented the unsupervised machine-learning approaches in the analysis of gene expression measurements. The following preprocessing steps were performed for further analysis: the background correction, normalization, and the quantitative evaluation of data quality. The next step was to find the differentially expressed gene from the preprocessed data using clustering. Then, we linked these observations for functional annotation in order to quantify the genes that share similar biological processes.

- The extraction of simple features was shown to be superior to more complicated feature-extraction pipelines without compromising the desired prediction accuracy of the classifier. Moreover, in model selection, the performance of the Bayesian minimum mean square error estimator (BEE) outperformed the conventional approaches for high-dimensional features with a small number of samples. We observed that the stability of the feature selectors oscillated and the selection of similar feature subsets was subject to the error estimator and the sample size setting. The experimental

results showed that BEE was more effective than its alternatives in terms of stability selection. Despite considering only flow cytometry data, we expect the relevance of our approach also to be applicable to other kinds of biomedical data.

- Feature-selection and feature-ranking approaches were shown to be effective in identifying the supporting pathways related to cyclic di-guanosine monophosphate (c-di-GMP) signaling proteins. The results corresponded to the hypotheses listed in Table 1 of **Publication IV**.

- We introduced a novel accuracy metric for binary classification tasks using the aforementioned BEE approach. The BEE was already shown to be computationally faster and to have a smaller error rate than the conventional approaches, such as resubsampling and cross-validation. Nevertheless, error estimation is one of the key tools for model selection, so alternative accuracy metrics, such as receiver operating characteristic curve is a more appropriate choice, as the latter can both consider the imbalanced distribution of the classes in the data and measure the ranking quality of the classifier. In this context, we derived a closed-form expression of the receiver operating characteristic metric for a two-class linear classifier with the Gaussian distribution and a common covariance matrix for both classes. The results with both synthetic and real-world data outperformed the other alternatives if the prior assumptions were not violated. The proposed approach had a tendency to overestimate the area under the receiver operating characteristic curve, yet its smaller variance compensated for the bias and resulted in an accurate estimate.

- We observed the randomness in the cross-validation approach, which was used for both model selection and model assessment. Cross-validation was a perfect choice when the number of observations was moderate (for example, more than 30). On the other hand, with a smaller number of observations (for example, less than 15), it was impractical in train-test spilt operation for K-fold cross-validation. Even leave-one-out cross-validation was prone to high variance. A lower number of observations with high-dimensional features is a typical scenario in biology, so an ideal method would distill only the essential features while producing reliable and well-generalized results with this typical scenario.

These aforementioned findings are associated with the research questions raised in Chapter 1. According to this study,

- No general machine-learning method can be found to suit all kinds of biological problems. Nevertheless, a regularization strategy can fit a sparse model which eventually reduces the dimensionality for high-dimensional data. This also makes the regularization approach a basis for automatic feature selection. These findings are presented in **Publications I**, **II**, **III**, and **IV**.

- The findings and observations from **Publication III** show the randomness in feature-selection ability. This encourages the use of the alternative Bayesian approach. Additionally, the observations from **Publications I-IV** have shown that resubsampling and/or cross-validation approaches are not suitable for $p \gg n$, especially if n is quite small. This motivated us to introduce a new alternative criterion for accuracy assessment and has contributed to the method proposed in **Publication V**.

To summarize, the relevance of machine-learning algorithms in biological and biomedical data can provide some interesting possibilities. These possibilities can improve the predictive capabilities of models learned from previously observed data in order to characterize unobserved data. Moreover, computational modeling and analysis can provide new and useful insights into a number of challenging questions in the field of biology. More research is still required to adapt these methodologies for new applications. Moreover, the success of these methodologies depends greatly on the quality and quantity of the data, which is a prerequisite for a predictive model to be successful.

Bibliography

[1] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson, "Big data: Astronomical or genomical?" *PLOS Biology*, vol. 13, no. 7, pp. 1–11, 07 2015. [Online]. Available: https://doi.org/10.1371/journal.pbio.1002195

[2] "Top 15 valuable facebook statistics," https://zephoria.com/top-15-valuable-facebook-statistics/, accessed: 2018-08-28.

[3] A. Szalay and J. Gray, "2020 computing: Science in an exponential world," *Nature*, vol. 440, no. 7083, p. 413, 2006.

[4] J. D. Watson, F. H. Crick *et al.*, "Molecular structure of nucleic acids," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.

[5] E. Birney, "The making of encode: lessons for big-data projects," *Nature*, vol. 489, no. 7414, p. 49, 2012.

[6] D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre *et al.*, "Big data: The future of biocuration," *Nature*, vol. 455, no. 7209, p. 47, 2008.

[7] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson, "Big data: astronomical or genomical?" *PLoS biology*, vol. 13, no. 7, p. e1002195, 2015.

[8] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.

[9] B. Efron and R. Tibshirani, "Statistical data analysis in the computer age," *Science*, vol. 253, no. 5018, pp. 390–395, 1991.

[10] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.

[11] P. Hall, J. S. Marron, and A. Neeman, "Geometric representation of high dimension, low sample size data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 3, pp. 427–444, 2005.

[12] H. M. Shapiro, *Practical flow cytometry*. John Wiley & Sons, 2005.

[13] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[14] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[16] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[17] S. Okser, T. Pahikkala, A. Airola, T. Salakoski, S. Ripatti, and T. Aittokallio, "Regularized machine learning in the genetic prediction of complex traits," *PLOS Genetics*, vol. 10, no. 11, p. e1004754, November 2014.

[18] Z. Guo, C. Li, L. Song, and L. V. Wang, "Compressed sensing in photoacoustic tomography in vivo," *Journal of biomedical optics*, vol. 15, no. 2, p. 021311, 2010.

[19] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan, "Learning The Invisible: A Hybrid Deep Learning-Shearlet Framework for Limited Angle Computed Tomography," *ArXiv e-prints*, Nov. 2018.

[20] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[21] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler, "Hidden markov models in computational biology: Applications to protein modeling," *Journal of Molecular Biology*, vol. 235, no. 5, pp. 1501–1531, 1994.

[22] A. Krogh, I. S. Mian, and D. Haussler, "A hidden markov model that finds genes in e. coli dna," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4768–4778, 1994.

[23] A. Duhamel, M. Nuttens, P. Devos, M. Picavet, and R. Beuscart, "A preprocessing method for improving data mining techniques. application to a large medical diabetes database." *Studies in health technology and informatics*, vol. 95, pp. 269–274, 2003.

[24] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 363–392, 2005.

[25] A. Bjorck, *Numerical methods for least squares problems*. Siam, 1996, vol. 51.

[26] D. Donoho, H. Kakavand, and J. Mammen, "The simplest solution to an underdetermined system of linear equations," in *2006 IEEE International Symposium on Information Theory*. IEEE, 2006, pp. 1924–1928.

[27] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4813–4820, 2008.

[28] I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider, and M. Uhr, "Competitive baseline methods set new standards for the nips 2003 feature selection benchmark," *Pattern recognition letters*, vol. 28, no. 12, pp. 1438–1444, 2007.

[29] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of human genetics*, vol. 7, no. 2, pp. 179–188, 1936.

[30] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[31] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[32] D. P. Bertsekas, A. Nedi, A. E. Ozdaglar *et al.*, "Convex analysis and optimization," 2003.

[33] S. M. Stigler, "Gauss and the invention of least squares," *Ann. Statist.*, vol. 9, no. 3, pp. 465–474, 05 1981.

[34] I. Markovsky and K. Usevich, *Low rank approximation*. Springer, 2012.

[35] E. Levin, N. Tishby, and S. A. Solla, "A statistical approach to learning and generalization in layered neural networks," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1568–1574, 1990.

[36] Š. Raudys and D. M. Young, "Results in statistical discriminant analysis: A review of the former soviet union literature," *Journal of Multivariate Analysis*, vol. 89, no. 1, pp. 1–35, 2004.

[37] D. Haussler, M. Kearns, H. S. Seung, and N. Tishby, "Rigorous learning curve bounds from statistical mechanics," *Machine Learning*, vol. 25, no. 2-3, pp. 195–236, 1996.

[38] S. Portnoy *et al.*, "Asymptotic behavior of m-estimators of p regression parameters when p2/n is large. i. consistency," *The Annals of Statistics*, vol. 12, no. 4, pp. 1298–1309, 1984.

[39] S. Portnoy, "Asymptotic behavior of m estimators of p regression parameters when p2/n is large; ii. normal approximation," *The Annals of Statistics*, pp. 1403–1417, 1985.

[40] A. B. Tsybakov, "Optimal rates of aggregation," in *Learning Theory and Kernel Machines*. Springer, 2003, pp. 303–313.

[41] C. R. Rao and V. Varadarajan, "Discrimination of gaussian processes," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 303–330, 1963.

[42] J. Fan and R. Li, "Statistical challenges with high dimensionality: Feature selection in knowledge discovery," *arXiv preprint math/0602133*, 2006.

[43] P. Domingos, "A unified bias-variance decomposition," in *Proceedings of 17th International Conference on Machine Learning*, 2000, pp. 231–238.

[44] R. E. Bellman, *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[45] N. Hoque, D. Bhattacharyya, and J. K. Kalita, "Mifs-nd: a mutual information-based feature selection method," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6371–6385, 2014.

[46] H. Huttunen, "Deep neural networks: A signal processing perspective," in *Handbook of Signal Processing Systems*. Springer, 2019, pp. 133–163.

[47] A. R. Webb, *Statistical pattern recognition.* John Wiley & Sons, 2003.

[48] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis.* John Wiley & Sons, 2012, vol. 821.

[49] P. Flach, *Machine learning: the art and science of algorithms that make sense of data.* Cambridge University Press, 2012.

[50] N. H. Bingham and J. M. Fry, *Regression: Linear models in statistics.* Springer Science & Business Media, 2010.

[51] R. L. Plackett, "Some theorems in least squares," *Biometrika*, vol. 37, no. 1/2, pp. 149–157, 1950.

[52] A. Birnbaum, "The neyman-pearson theory as decision theory, and as inference theory; with a criticism of the lindley-savage argument for bayesian theory," *Synthese*, vol. 36, no. 1, pp. 19–49, 1977.

[53] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242, 1958.

[54] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, no. 1-2, pp. 167–179, 1967.

[55] J. S. Cramer, "The origins of logistic regression," 2002.

[56] R. D. B., *Iteratively Reweighted Least Squares.* American Cancer Society, 2006.

[57] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks*, vol. 2, no. 2004, p. 41, 2004.

[58] S. Marsland, *Machine learning: an algorithmic perspective.* CRC press, 2015.

[59] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees.* CRC press, 1984.

[60] R. Timofeev, "Classification and regression trees (cart) theory and applications," *Humboldt University, Berlin*, 2004.

[61] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Annals of statistics*, pp. 1651–1686, 1998.

[62] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[63] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[64] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[65] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC bioinformatics*, vol. 8, no. 1, p. 25, 2007.

[66] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC bioinformatics*, vol. 9, no. 1, p. 307, 2008.

[67] E. Tuv, A. Borisov, G. Runger, and K. Torkkola, "Feature selection with ensembles, artificial variables, and redundancy elimination," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1341–1366, 2009.

[68] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[69] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[70] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[71] E. Naghieh and Y. Peng, "Microarray gene expression data mining: clustering analysis review," *Department of Computing*, 2009.

[72] F. Azuaje and N. Bolshakova, "Clustering genomic expression data: design and evaluation principles," in *A Practical Approach to Microarray Data Analysis*. Springer, 2003, pp. 230–245.

[73] M. M. Babu, "Introduction to microarray data analysis," *Computational genomics: Theory and application*, vol. 17, no. 6, pp. 225–49, 2004.

[74] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Transactions on knowledge and data engineering*, vol. 16, no. 11, pp. 1370–1386, 2004.

[75] A. Thalamuthu, I. Mukhopadhyay, X. Zheng, and G. C. Tseng, "Evaluation and comparison of gene clustering methods in microarray analysis," *Bioinformatics*, vol. 22, no. 19, pp. 2405–2412, 2006.

[76] S. I. Kabanikhin, "Definitions and examples of inverse and ill-posed problems," *Journal of Inverse and Ill-Posed Problems*, vol. 16, no. 4, pp. 317–357, 2008.

[77] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[78] A. Tikhonov and V. Y. Arsenin, *Methods for solving ill-posed problems*. John Wiley and Sons, Inc, 1977.

[79] A. X. Zheng, M. I. Jordan, B. Liblit, and A. Aiken, "Statistical debugging of sampled programs," in *Advances in Neural Information Processing Systems*, 2004, pp. 603–610.

[80] M. Schmidt, "Least squares optimization with l1-norm regularization," *CS542B Project Report*, pp. 14–18, 2005.

[81] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[82] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, 1993.

[83] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[84] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *European Conference on Machine Learning*. Springer, 2007, pp. 286–297.

[85] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[86] T. Hastie, *The elements of statistical learning data mining, inference, and prediction*, R. Tibshirani and J. Friedman, Eds. New York: Springer, 2009, vol. 2nd ed.

[87] J. A. Nelder and R. J. Baker, *Generalized linear models*. Wiley Online Library, 1972.

[88] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[89] O. D. Trier, A. K. Jain, T. Taxt *et al.*, "Feature extraction methods for character recognition-a survey," *Pattern recognition*, vol. 29, no. 4, pp. 641–662, 1996.

[90] P. A. Devijver and J. Kittler, *Pattern recognition: A statistical approach*. Prentice hall, 1982.

[91] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[92] K. Fukunaga, *Introduction to statistical pattern recognition*. Elsevier, 2013.

[93] H. H. Yang and J. Moody, "Data visualization and feature selection: New algorithms for nongaussian data," in *Advances in Neural Information Processing Systems*, 2000, pp. 687–693.

[94] M. Girolami, A. Cichocki, and S.-I. Amari, "A common neural-network model for unsupervised exploratory data analysis and independent component analysis," *IEEE transactions on neural networks*, vol. 9, no. 6, pp. 1495–1501, 1998.

[95] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2015, pp. 1200–1205.

[96] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[97] S. Ma and J. Huang, "Penalized feature selection and classification in bioinformatics," *Briefings in bioinformatics*, vol. 9, no. 5, pp. 392–403, 2008.

[98] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[99] P. M. Chelvan and K. Perumal, "A comparative analysis of feature selection stability measures," in *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, May 2017, pp. 124–128.

[100] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and information systems*, vol. 12, no. 1, pp. 95–116, 2007.

[101] E. R. Dougherty, C. Sima, B. Hanczar, U. M. Braga-Neto *et al.*, "Performance of error estimators for classification," *Current Bioinformatics*, vol. 5, no. 1, pp. 53–67, 2010.

[102] A. M. Molinaro, R. Simon, and R. M. Pfeiffer, "Prediction error estimation: a comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, pp. 3301–3307, 2005.

[103] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2079–2107, 2010.

[104] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27 – 38, 2009.

[105] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427 – 437, 2009.

[106] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. ACM press New York, 1999, vol. 463.

[107] W. P. Tanner Jr and J. A. Swets, "A decision-making theory of visual detection." *Psychological review*, vol. 61, no. 6, p. 401, 1954.

[108] K. H. Zou, "Receiver operating characteristic (roc) literature research," 2002.

[109] F. Gorunescu, *Data Mining: Concepts, models and techniques*. Springer Science & Business Media, 2011, vol. 12.

[110] L. Gonçalves, A. Subtil, M. R. Oliveira, and P. Bermudez, "Roc curve estimation: An overview," *REVSTAT–Statistical Journal*, vol. 12, no. 1, pp. 1–20, 2014.

[111] P. A. Flach and S. Wu, "Repairing concavities in roc curves." in *IJCAI*, 2005, pp. 702–707.

[112] D. Mossman, "Three-way rocs," *Medical Decision Making*, vol. 19, no. 1, pp. 78–89, 1999.

[113] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.

[114] T. C. Landgrebe and R. P. Duin, "Efficient multiclass roc approximation by decomposition via confusion matrix perturbation analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 5, pp. 810–822, 2008.

[115] M. S. Pepe, *The statistical evaluation of medical tests for classification and prediction*. Medicine, 2003.

[116] D. J. Hand, "Measuring classifier performance: a coherent alternative to the area under the roc curve," *Machine learning*, vol. 77, no. 1, pp. 103–123, 2009.

[117] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[118] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[119] L. E. Bantis and Z. Feng, "Comparison of two correlated roc curves at a given specificity or sensitivity level," *Statistics in medicine*, vol. 35, no. 24, pp. 4352–4367, 2016.

[120] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[121] K. O. Hajian-Tilaki, J. A. Hanley, L. Joseph, and J.-P. Collet, "Extension of receiver operating characteristic analysis to data concerning multiple signal detection tasks," *Academic radiology*, vol. 4, no. 3, pp. 222–229, 1997.

[122] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *arXiv preprint arXiv:1809.03006*, 2018.

[123] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthey Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.

[124] U. M. B. Neto and E. R. Dougherty, *Error estimation for pattern recognition*. John Wiley & Sons, 2015.

[125] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[126] J. Neyman, "On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection," *Journal of the Royal Statistical Society*, vol. 97, no. 4, pp. 558–625, 1934.

[127] L. Dalton and E. R. Dougherty, "Bayesian minimum mean-square error estimation for classification error—–part I: Definition and the bayesian MMSE error estimator for discrete classification," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 115–129, 2011.

[128] L. Dalton and E. R. Dougherty, "Bayesian minimum mean-square error estimation for classification error—–part II: The bayesian MMSE error estimator for linear classification of gaussian distributions," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 130––144, 2011.

[129] H. Huttunen, T. Manninen, and J. Tohka, "Bayesian error estimation and model selection in sparse logistic regression," in *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*. IEEE, 2013, pp. 1–6.

[130] H. Huttunen and J. Tohka, "Model selection for linear classifiers using bayesian error estimation," *Pattern Recognition*, vol. 48, no. 11, pp. 3739–3748, 2015.

[131] L. A. Dalton, "Optimal roc-based classification and performance analysis under bayesian uncertainty models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 4, pp. 719–729, 2016.

[132] O. V. Demler, M. J. Pencina, and R. B. D'Agostino Sr, "Equivalence of improvement in area under roc curve and linear discriminant analysis coefficient under assumption of normality," *Statistics in medicine*, vol. 30, no. 12, pp. 1410–1418, 2011.

[133] J. P. Arnold, *Origin and History of Beer Brewing: From Prehistoric Times to the Beginning of Brewing Science and Technology*. BeerBooks Cleveland, OH, 2005.

[134] I. Orhan, *Biotechnological production of plant secondary metabolites*. Bentham science publishers, 2012.

[135] A. Fleming, "On the antibacterial action of cultures of a penicillium, with special reference to their use in the isolation of b. influenzae," *British journal of experimental pathology*, vol. 10, no. 3, p. 226, 1929.

[136] A. L. Demain, "From natural products discovery to commercialization: a success story," *Journal of Industrial Microbiology and Biotechnology*, vol. 33, no. 7, pp. 486–495, 2006.

[137] M. F. Cantley, "The regulation of modern biotechnology: a historical and european perspective: a case study in how societies cope with new knowledge in the last quarter of the twentieth century," *Biotechnology Set, Second Edition*, pp. 505–681, 2008.

[138] J. S. Alford, "Bioprocess control: Advances and challenges," *Computers & Chemical Engineering*, vol. 30, no. 10-12, pp. 1464–1475, 2006.

[139] X. Y. Lawrence, "Pharmaceutical quality by design: product and process development, understanding, and control," *Pharmaceutical research*, vol. 25, no. 4, pp. 781–791, 2008.

[140] Food, D. Administration *et al.*, "Guidance for industry, pat-a framework for innovative pharmaceutical development, manufacturing and quality assurance," *http://www. fda. gov/cder/guidance/published. html*, 2004.

[141] R. Mangayil, M. Karp, and V. Santala, "Bioconversion of crude glycerol from biodiesel production to hydrogen," *international journal of hydrogen energy*, vol. 37, no. 17, pp. 12 198–12 204, 2012.

[142] R. Mangayil, T. Aho, M. Karp, and V. Santala, "Improved bioconversion of crude glycerol to hydrogen by statistical optimization of media components," *Renewable Energy*, vol. 75, pp. 583–589, 2015.

[143] R. L. Plackett and J. P. Burman, "The design of optimum multifactorial experiments," *Biometrika*, vol. 33, no. 4, pp. 305–325, 1946.

[144] G. E. Box and D. W. Behnken, "Some new three level designs for the study of quantitative variables," *Technometrics*, vol. 2, no. 4, pp. 455–475, 1960.

[145] A. E. Hoerl, "Optimum solution of many variables equations," *Chemical Engineering Progress*, vol. 55, no. 11, pp. 69–78, 1959.

[146] D. C. Montgomery, *Design and analysis of experiments*. John wiley & sons, 2017.

[147] C.-F. Mandenius and A. Brundin, "Bioprocess optimization using design-of-experiments methodology," *Biotechnology progress*, vol. 24, no. 6, pp. 1191–1203, 2008.

[148] T. Brown, *Genomes 2nd edition*. Garland Science, 2002.

[149] M. Campbell and I. Geis, *Biochemistry*, ser. Saunders golden sunburst series. Saunders College Pub., 1995.

[150] R. Govindarajan, J. Duraiyan, K. Kaliyappan, and M. Palanisamy, "Microarray and its applications," *Journal of pharmacy & bioallied sciences*, vol. 4, no. Suppl 2, p. S310, 2012.

[151] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS letters*, vol. 480, no. 1, pp. 17–24, 2000.

[152] A. Laiho *et al.*, "Data analysis tools and methods for dna microarray and high-throughput sequencing data," 2016.

[153] K. Aas, "Microarray data mining: A survey," *NR Note, SAMBA, Norwegian Computing Center*, 2001.

[154] L. M. Smoot, J. C. Smoot, M. R. Graham, G. A. Somerville, D. E. Sturdevant, C. A. L. Migliaccio, G. L. Sylva, and J. M. Musser, "Global differential gene expression in response to growth temperature alteration in group a streptococcus," *Proceedings of the National Academy of Sciences*, vol. 98, no. 18, pp. 10 416–10 421, 2001.

[155] J. J. Mekalanos, "Environmental signals controlling expression of virulence determinants in bacteria." *Journal of bacteriology*, vol. 174, no. 1, p. 1, 1992.

[156] S. Huehn, C. Eichhorn, S. Urmersbach, J. Breidenbach, S. Bechlars, N. Bier, T. Alter, E. Bartelt, C. Frank, B. Oberheitmann *et al.*, "Pathogenic vibrios in environmental, seafood and clinical sources in germany," *International Journal of Medical Microbiology*, vol. 304, no. 7, pp. 843–850, 2014.

[157] T. Maier, M. Güell, and L. Serrano, "Correlation of mrna and protein in complex biological samples," *FEBS letters*, vol. 583, no. 24, pp. 3966–3973, 2009.

[158] I. Hovatta, K. Kimppa, A. Lehmussola, T. Pasanen, J. Saarela, I. Saarikko, J. Saharinen, P. Tiikkainen, T. Toivanen, M. Tolvanen *et al.*, "Dna microarray data analysis," *CSC, 2nd edn., Scientific Computing Ltd*, 2005.

[159] M. Brown and C. Wittwer, "Flow cytometry: principles and clinical applications in hematology," *Clinical chemistry*, vol. 46, no. 8, pp. 1221–1229, 2000.

[160] G. Henel and J. L. Schmitz, "Basic theory and clinical applications of flow cytometry," *Laboratory Medicine*, vol. 38, no. 7, pp. 428–436, 2007.

[161] T. G. Willis and M. J. Dyer, "The role of immunoglobulin translocations in the pathogenesis of b-cell malignancies," *Blood*, vol. 96, no. 3, pp. 808–822, 2000.

[162] M. J. Borowitz, K. Lynn Guenther, K. E. Shults, and G. T. Stelzer, "Immunophenotyping of acute leukemia by flow cytometric analysis: use of cd45 and right-angle light scatter to gate on leukemic blasts in three-color analysis," *American journal of clinical pathology*, vol. 100, no. 5, pp. 534–540, 1993.

[163] S. P. Perfetto, P. K. Chattopadhyay, and M. Roederer, "Seventeen-colour flow cytometry: unravelling the immune system," *Nature Reviews Immunology*, vol. 4, no. 8, p. 648, 2004.

[164] G. Lee, "Machine learning for flow cytometry data analysis," Ph.D. dissertation, University of Michigan, 2011.

[165] E. Lugli, M. Roederer, and A. Cossarizza, "Data analysis in flow cytometry: the future just started," *Cytometry Part A*, vol. 77, no. 7, pp. 705–713, 2010.

[166] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, R. H. Scheuermann, F. Consortium, D. Consortium *et al.*, "Critical assessment of automated flow cytometry data analysis techniques," *Nature methods*, vol. 10, no. 3, p. 228, 2013.

[167] R. M. Brown, "Cellulose structure and biosynthesis: What is in store for the 21st century?" *Journal of Polymer Science Part A: Polymer Chemistry*, vol. 42, no. 3, pp. 487–495, 2004.

[168] P. Gillis, R. Mark, and R.-C. Tang, "Elastic stiffness of crystalline cellulose in the folded-chain solid state," *Journal of Materials Science*, vol. 4, no. 11, pp. 1003–1007, 1969.

[169] C. T. Brett, "Cellulose microfibrils in plants: biosynthesis, deposition, and integration into the cell wall," *International review of cytology*, vol. 199, pp. 161–199, 2000.

[170] C. Somerville, "Cellulose synthesis in higher plants," *Annu.Rev.Cell Dev.Biol.*, vol. 22, pp. 53–78, 2006.

[171] S. Vitta and V. Thiruvengadam, "Multifunctional bacterial cellulose and nanoparticle-embedded composites," *Current Science(Bangalore)*, vol. 102, no. 10, pp. 1398–1405, 2012.

[172] E. Canale-Parola, "Biology of the sugar-fermenting sarcinae," *Bacteriological Reviews*, vol. 34, no. 1, pp. 82–97, Mar 1970, lR: 20131121; JID: 0370620; 0 (Culture Media); 9004-34-6 (Cellulose); IY9XDZ35W2 (Glucose); RF: 100; OID: NLM: PMC378349; ppublish.

[173] S. Park, J. O. Baker, M. E. Himmel, P. A. Parilla, and D. K. Johnson, "Research cellulose crystallinity index: measurement techniques and their impact on interpreting cellulase performance," *Biotechnol Biofuels*, vol. 3, no. 10, 2010.

[174] P. Ross, R. Mayer, and M. Benziman, "Cellulose biosynthesis and function in bacteria," *Microbiological reviews*, vol. 55, no. 1, pp. 35–58, Mar 1991, lR: 20131002; JID: 7806086; 9004-34-6 (Cellulose); RF: 159; OID: NLM: PMC372800; ppublish.

[175] R. H. Atalla and D. L. Vanderhart, "Native cellulose: a composite of two distinct crystalline forms," *Science (New York, N.Y.)*, vol. 223, no. 4633, pp. 283–285, Jan 20 1984, jID: 0404511; ppublish.

[176] D. N. Hon, "Cellulose: a random walk along its historical path," *Cellulose*, vol. 1, no. 1, pp. 1–25, 1994.

[177] D. Klemm, F. Kramer, S. Moritz, T. Lindström, M. Ankerfors, D. Gray, and A. Dorris, "Nanocelluloses: A new family of nature-based materials," *Angewandte Chemie International Edition*, vol. 50, no. 24, pp. 5438–5466, 2011.

[178] K. Lee, G. Buldum, A. Mantalaris, and A. Bismarck, "More than meets the eye in bacterial cellulose: Biosynthesis, bioprocessing, and applications in advanced fiber composites," *Macromolecular bioscience*, vol. 14, no. 1, pp. 10–32, 2014.

[179] Y. Huang, C. Zhu, J. Yang, Y. Nie, C. Chen, and D. Sun, "Recent advances in bacterial cellulose," *Cellulose*, vol. 21, no. 1, pp. 1–30, 2014.

[180] M. Iguchi, S. Yamanaka, and A. Budhiono, "Bacterial cellulose a masterpiece of natureś arts," *Journal of Materials Science*, vol. 35, no. 2, pp. 261–270, 2000.

[181] E. E. Brown and M.-P. G. Laborie, "Bioengineering bacterial cellulose/poly (ethylene oxide) nanocomposites," *Biomacromolecules*, vol. 8, no. 10, pp. 3074–3081, 2007.

[182] S. Yamanaka, K. Watanabe, N. Kitamura, M. Iguchi, S. Mitsuhashi, Y. Nishi, and M. Uryu, "The structure and mechanical properties of sheets prepared from bacterial cellulose," *Journal of Materials Science*, vol. 24, no. 9, pp. 3141–3145, 1989.

[183] W. Wan, J. Hutter, L. Millon, and G. Guhados, "Bacterial cellulose and its nanocomposites for biomedical applications," in *ACS symposium series*, vol. 938. Oxford University Press, 2006, pp. 221–241.

[184] H. S. Barud, C. Barrios, T. Regiani, R. F. Marques, M. Verelst, J. Dexpert-Ghys, Y. Messaddeq, and S. J. Ribeiro, "Self-supported silver nanoparticles containing bacterial cellulose membranes," *Materials Science and Engineering: C*, vol. 28, no. 4, pp. 515–518, 2008.

[185] I. Siró and D. Plackett, "Microfibrillated cellulose and new nanocomposite materials: a review," *Cellulose*, vol. 17, no. 3, pp. 459–494, 2010.

[186] D. Klemm, D. Schumann, F. Kramer, N. Heßler, M. Hornung, H.-P. Schmauder, and S. Marsch, *Nanocelluloses as innovative polymers in research and application*, ser. Polysaccharides Ii. Springer, 2006, pp. 49–96.

[187] W. Czaja, D. Romanovicz, and R. Malcolm Brown, "Structural investigations of microbial cellulose produced in stationary and agitated culture," *Cellulose*, vol. 11, no. 3-4, pp. 403–411, 2004.

[188] J. Brown, R. M., J. H. Willison, and C. L. Richardson, "Cellulose biosynthesis in acetobacter xylinum: visualization of the site of synthesis and direct measurement of the in vivo process," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 73, no. 12, pp. 4565–4569, Dec 1976, lR: 20091118; JID: 7505876; 9004-34-6 (Cellulose); OID: NLM: PMC431544; ppublish.

[189] U. Römling and M. Y. Galperin, "Bacterial cellulose biosynthesis: diversity of operons, subunits, products, and functions," *Trends in microbiology*, vol. 23, no. 9, pp. 545–557, 2015.

[190] M. Valentini and A. Filloux, "Biofilms and c-di-gmp signaling: lessons from pseudomonas aeruginosa and other bacteria," *Journal of Biological Chemistry*, pp. jbc–R115, 2016.

[191] D. Van De Ville and S.-W. Lee, "Brain decoding: Opportunities and challenges for pattern recognition," *Pattern Recognition*, vol. 45, no. 6, pp. 2033–2034, 2012.

[192] P. M. Rasmussen, L. K. Hansen, K. H. Madsen, N. W. Churchill, and S. C. Strother, "Model sparsity and brain pattern interpretation of classification models in neuroimaging," *Pattern Recognition*, vol. 45, no. 6, pp. 2085–2100, 2012.

[193] T. Naselaris, K. N. Kay, S. Nishimoto, and J. L. Gallant, "Encoding and decoding in fmri," *Neuroimage*, vol. 56, no. 2, pp. 400–410, 2011.

[194] F. Pereira, T. Mitchell, and M. Botvinick, "Machine learning classifiers and fmri: a tutorial overview," *Neuroimage*, vol. 45, no. 1, pp. S199–S209, 2009.

[195] A. J. O'Toole, F. Jiang, H. Abdi, N. Pénard, J. P. Dunlop, and M. A. Parent, "Theoretical, statistical, and practical perspectives on pattern-based classification approaches to the analysis of functional neuroimaging data," *Journal of cognitive neuroscience*, vol. 19, no. 11, pp. 1735–1752, 2007.

[196] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Müller, "Introduction to machine learning for brain imaging," *NeuroImage*, vol. 56, no. 2, pp. 387 – 399, 2011, multivariate Decoding and Brain Reading.

[197] H. Huttunen, T. Manninen, J.-P. Kauppi, and J. Tohka, "Mind reading with regularized multinomial logistic regression," *Machine vision and applications*, vol. 24, no. 6, pp. 1311–1325, 2013.

[198] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer, 2005, pp. 397–420.

[199] M. Biehl, K. Bunte, and P. Schneider, "Analysis of flow cytometry data by matrix relevance learning vector quantization." *PLoS One*, vol. 8, no. 3, p. e59401, 2013.

[200] N. Aghaeepour, G. Finak, T. F. Consortium, T. D. Consortium, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, and R. H. Scheuermann, "Critical assessment of automated flow cytometry data analysis techniques," *Nat Meth*, vol. 10, pp. 228–237, 2013.

[201] T. Manninen, H. Huttunen, P. Ruusuvuori, and M. Nykter, "Leukemia prediction using sparse logistic regression." *PLoS One*, vol. 8, no. 8, p. e72932, 2013.

[202] D. Dernoncourt, B. Hanczar, and J.-D. Zucker, "Analysis of feature selection stability on high dimension and small sample data," *Computational statistics & data analysis*, vol. 71, pp. 681–693, 2014.

Publications

Publication I

Syeda Sakira Hassan, Muhammad Farhan, Rahul Mangayil, Heikki Huttunen, and Tommi Aho. "Bioprocess data mining using regularized regression and random forests", *BMC Systems Biology*, vol 7, no. Supp 1, pp. S(5), Aug. 2013.

Hassan et al. BMC Systems Biology 2013, 7(Suppl 1):S5
http://www.biomedcentral.com/1752-0509/7/S1/S5

BMC Systems Biology

RESEARCH Open Access

Bioprocess data mining using regularized regression and random forests

Syeda Sakira Hassan[1*†], Muhammad Farhan[1†], Rahul Mangayil[2], Heikki Huttunen[1], Tommi Aho[2]

From 10th International Workshop on Computational Systems Biology
Tampere, Finland. 10-12 June 2013

Abstract

Background: In bioprocess development, the needs of data analysis include (1) getting overview to existing data sets, (2) identifying primary control parameters, (3) determining a useful control direction, and (4) planning future experiments. In particular, the integration of multiple data sets causes that these needs cannot be properly addressed by regression models that assume linear input-output relationship or unimodality of the response function. Regularized regression and random forests, on the other hand, have several properties that may appear important in this context. They are capable, e.g., in handling small number of samples with respect to the number of variables, feature selection, and the visualization of response surfaces in order to present the prediction results in an illustrative way.

Results: In this work, the applicability of regularized regression (Lasso) and random forests (RF) in bioprocess data mining was examined, and their performance was benchmarked against multiple linear regression. As an example, we used data from a culture media optimization study for microbial hydrogen production. All the three methods were capable in providing a significant model when the five variables of the culture media optimization were linearly included in modeling. However, multiple linear regression failed when also the multiplications and squares of the variables were included in modeling. In this case, the modeling was still successful with Lasso (correlation between the observed and predicted yield was 0.69) and RF (0.91).

Conclusion: We found that both regularized regression and random forests were able to produce feasible models, and the latter was efficient in capturing the non-linearity in the data. In this kind of a data mining task of bioprocess data, both methods outperform multiple linear regression.

Background

Industrial biotechnology exploits processes that use living cells, for instance yeast and various bacteria, to produce products like fine chemicals, active pharmaceutical ingredients, enzymes, and biofuels. The use of living material in manufacturing processes makes the processes challenging to develop and control. Because of the complexity of these tasks, computational modeling and data analysis are used to improve the yield, reproducibility and robustness in bioprocesses. On the other hand, the regulatory demands on pharmaceutical manufacturing processes are increasing and, for example, the United States Food and Drug Administration emphasize the importance of model-aided process development in its process analytical technology (PAT) initiative [1]. One of the important steps in process development is maximizing the product yield. In practice, the process optimization includes (1) identifying the process parameters that have most impact to the product yield and, (2) determining their optimal values. This data analysis task includes few features that are specific to the application area. For example, the number of process parameters (predictors) may be large with respect to the number of samples, the predictors may contain either numerical or categorical values, the datasets may contain

* Correspondence: sakira.hassan@tut.fi
† Contributed equally
[1]Department of Signal Processing, Tampere University of Technology, Tampere, P.O. Box 553, 33101, Finland
Full list of author information is available at the end of the article

missing values and, finally, the relationship among the predictors and product yield may be non-linear.

To build a model for data analysis requires selection of important features while leaving out the rest. Several feature selection methods have been proposed but the results tend to vary, as generalization of the solution is problematic. Typical issues are data redundancy, outliers and feature dependencies [2,3].

Methods

In this work, we have used three alternative approaches to model bioprocess data: multiple linear regression, regularized regression and random forests. The analyses were performed using MATLAB [4] and RF-ACE tool [5].

Multiple linear regression

In multiple linear regression, the response variable is modeled as a linear combination of multiple predictor variables. The general model can be expressed as

$$y = \beta_0 + a_1\beta_1 + a_2\beta_2 + a_3\beta_3 + \ldots + a_p\beta_p \quad (1)$$

where y is the response variable, and a_i and β_i (i = 1, ..., p) are the predictor variables and their coefficients, respectively. The intercept is represented by β_0. Alternatively, Equation (1) can be represented in vector notation by $\mathbf{y} = \mathbf{H}\theta$, where $\mathbf{H}$ is augmented predictor vector given as [$\mathbf{1}$ $\boldsymbol{a_1}$ $\boldsymbol{a_2}$ ... $\boldsymbol{a_p}$] and θ is the parameter vector.

In spite of being linear with respect to the predictor variables, multiple linear regression models fail to incorporate the underlying non-linear relationships, if it exists, between the predictors and the response variable. However, the model restricts only the coefficients to be linearly related, while the predictor variables can be non-linear. This gives a provision of including additional non-linearly transformed predictor variables in the linear regression modeling. The advantage of using such variables in regression analysis is that the non-linear behavior in data and interaction between different variables are incorporated while the model remains linear and easily interpretable. This is a typical procedure applied in traditional response surface modeling when constructing models with quadratic terms and interactions of terms. Increasing the number of parameters in this way, however, causes high-dimensional predictor vector which results in over-fitting and the loss of generality. Moreover, if the number of samples is small, increasing the parameter vector size by these transformations may cause rank deficiency or multicollinearity of the prediction vector. In such cases, standard regression modeling may either fail, rank deficiency may cause non-invertible matrix thus making parameter estimation difficult, or the estimates it gives for parameter vector are prone to give low prediction accuracy. Hence, regularization is a key process in solving such cases. It produces a sparse parameter vector and also shrinks the coefficients towards zero as well as towards each other [6].

Regularized regression

The research on sparse and regularized solutions has gained increasing interest during the last ten years [7]. This is partly due to advances in measurement technologies, e.g., in molecular biology, where high-throughput technologies allow simultaneous measurement of tens of thousands of variables. However, the measurements are expensive, so typically the number of data points is small. In the field of bioprocess development, the number of variables is not that large but yet enough to hinder the use of many standard data analysis methods. Conventional regression and classification methods are unable to process data with more predictor variables than samples (so called $p >> N$ problem). Regularization methods help in defining a unique solution in this ill-posed problem. These methods shrink some of the coefficients to zero. This not only helps in feature selection but also decreases the variance at the cost of a small increase in bias. However, this has the effect of improving the generalization of the estimate.

In regularized regression, a penalty on the size of the coefficients is added to the error function. Least absolute shrinkage and selection operator (LASSO) [3] is one such technique which uses the L_1 norm of the coefficients as the penalty term to produce *sparse* solutions, i.e., prediction models with several coefficients equal to zero. Since variables with zero coefficients are not used, this procedure essentially acts as an embedded feature selection.

From the description of Equation (1), the L_1 penalized coefficient vector for our linear model is defined as

$$\hat{\boldsymbol{\theta}} = ||\boldsymbol{\gamma} - \boldsymbol{H\theta}||_2^2 + \lambda||\boldsymbol{\theta}||_1 \quad (2)$$

where lambda (λ) is the regularization parameter, $||\theta||_1$ is the L_1-norm of the parameter vector. There exist efficient algorithms for finding solutions for different values of regularization parameters [3].

The result of the regularized regression is quite sensitive to the selection of the parameter λ. In order to appropriately assess the performance, the selection has to be done based on data. The usual approach is to estimate the performance with different λ using a cross-validation approach. Since we also use cross-validation for estimating the performance of the overall method (including the algorithm for selecting λ), this results in two nested cross-validation loops, one for model selection and one for error estimation. More specifically, the outer loop is used for estimating the performance for new data, while the inner loop is used for selection of λ.

Random forests

Decision trees have been studied for decades as a model for various prediction problems. The tree can be either a classification tree or a regression tree, and a common term including both is classification and regression tree (CART). A decision tree is a hierarchical structure, which decides the class (in classification) or the predicted output (regression) by hierarchically comparing feature values with a selected threshold, thus producing a hierarchy of if-then rules. Such combination of rules is most conveniently expressed as a tree, where each input feature comparison corresponds to a node in the tree. Eventually, the leaves of the tree describe the actual output value.

The decision trees can be learned from the data, and the usual approach is to add nodes using a top-down greedy algorithm. In essence, this means dividing the search space into rectangular regions according to the splitting points. The drawback of decision tree is that they are very prone to overlearning. This is one reason why regression trees have later been extended to random forests [8], whose prediction is obtained by averaging the outputs of a large number of regression trees. Due to averaging, random forests are tolerant to overlearning, a typical phenomenon in high-dimensional settings with small sample size, and have thus gained popularity in classification and regression tasks especially in the area of bioinformatics.

In our experiments, we use the RF-ACE implementation in [5]. This implementation is very fast and it takes advantage of the Random Forest with Artificial Ensembles (RF-ACE) algorithm, which enables both feature ranking and model construction. In our approach, a set of significant features was first selected from the experimental data using the RF-ACE tool. Then, a model was constructed using the given data.

Experimental data

In order to test our modeling methodology we examined a dataset produced in a study related to culture media optimization (unpublished data, Rahul Mangayil et al.). There, an enriched mixed microbial consortium was used in the bioconversion of crude glycerol to hydrogen, and the process was optimized in serum bottles by optimization of media components. The concentrations of five media components (NH_4Cl, K_2HPO_4, KH_2PO_4, $MgCl_2.6H_2O$, and KCl) were varied with the help of statistical design of experiments (Plackett-Burman, steepest ascent, Box-Behnken), and the resulting hydrogen production was measured (in mol-H_2/mol-glycerol). The data was modeled using first and second order polynomials in multiple linear regression. This data containing 35 samples is a typical data set produced during bioprocess modeling and optimization. Multiple linear regression is a useful tool for modeling the data from individual designs of the study but other methods are needed in order to model the entire data set at once.

Visualization and validation of models

In order to provide an overview to the models and the experimental data, visual representations were produced for the regularized regression model and the random forest model. Since visualization of the high dimensional variable space (five dimensions in our case study) is not feasible, the variables are visualized pair-wise. The values of remaining variables (three) are set in their average values calculated from the data. In addition, each model is assessed with *leave-one-out* (LOO) cross validation technique which estimates the accuracy of the predictions in an independent dataset.

Results and discussion

In our case study, we used multiple linear regression, regularized regression and random forests to predict the yield of hydrogen production. The performance of each method is evaluated by original dataset as well as transformed dataset with pairwise interactions and quadratic forms. Therefore, the original dataset contains 5 variables while the transformed dataset contains 20 variables.

Yield prediction using multiple linear regression

Multiple linear regression is used with and without nonlinearly transformed predictor variables to model the response variable. Without the transformed predictors, i.e., the simple model, the estimated correlation value (using the LOO cross-validation) was 0.65. However, using the transformed polynomial model the estimate for correlation decreased to a very low value of 0.012 and resulted in an insignificant model. This is mainly due to the aforementioned shortcomings of the multiple linear regression. It basically over-fits the model to the training samples and thus produces less accurate estimates for unseen data samples. Table S1 lists the model coefficients for the transformed polynomial regression model [see Additional file 1]. It can be noted that zero entries have been inserted to remove linearly dependent observations.

Yield prediction using regularized regression

First, we evaluated the simple model without the transformed variables. In this case, the parameter λ for the regularized regression is chosen by both manual selection and proper cross validation. In other words, we wanted to see if the results improve by manually selecting the lambda value optimally for each LOO cross validation fold. Although this is not possible in practical applications, it may give insight on the efficiency of

parameter selection using cross-validation with small sample size, and on the general applicability of a linear model for our problem.

As a result, the LOO correlation estimate becomes 0.85 with manual selection instead of 0.60 using proper cross-validation. The large gap between optimal and estimated correlation is at least in part due to the inaccuracy of the cross-validation type error estimators with small sample size; see, e.g., [9].

In the case of transformed polynomial regression model, the estimated value for correlation was found to be 0.69 which is higher than the case of the simple model. This clearly indicates the non-linear behavior of the original dataset. Table S1 shows the resulting coefficients in the constructed model where regularization has forced 5 out of 21 coefficients to zero [see Additional file 1]. Although, the same number of non-zero coefficients were obtained from the multiple linear regression as well but the main difference is the regularized coefficients. That is, the non-zero coefficients from regularized regression were also shrunk towards zero. This results in generalized models with higher overall prediction accuracy [3]. The yield predictions are visualized in Figure 1 as a response surface. In addition, the significant variables for the model and their corresponding coefficients are listed in Table 1.

Yield prediction using random forests

The RF-ACE tool [5] is used to build the random forests model. In our experiment, the type of the forest, the number of trees in the forest, and the fraction of randomly drawn features per node split are set to "RF", 20, and 10, respectively. All other parameters were kept to their default values. The results indicated that all variables were significant in the model. The yield predictions of the constructed model are visualized in Figure 2. In the accuracy examination, the RF-ACE model resulted in correlation of 0.88 (using LOO cross-validation). The capability of modeling non-linear relationships is the primary reason for high prediction accuracy in the constructed model. On the other hand, the model provided correlation value of 0.91 if the variable transformations

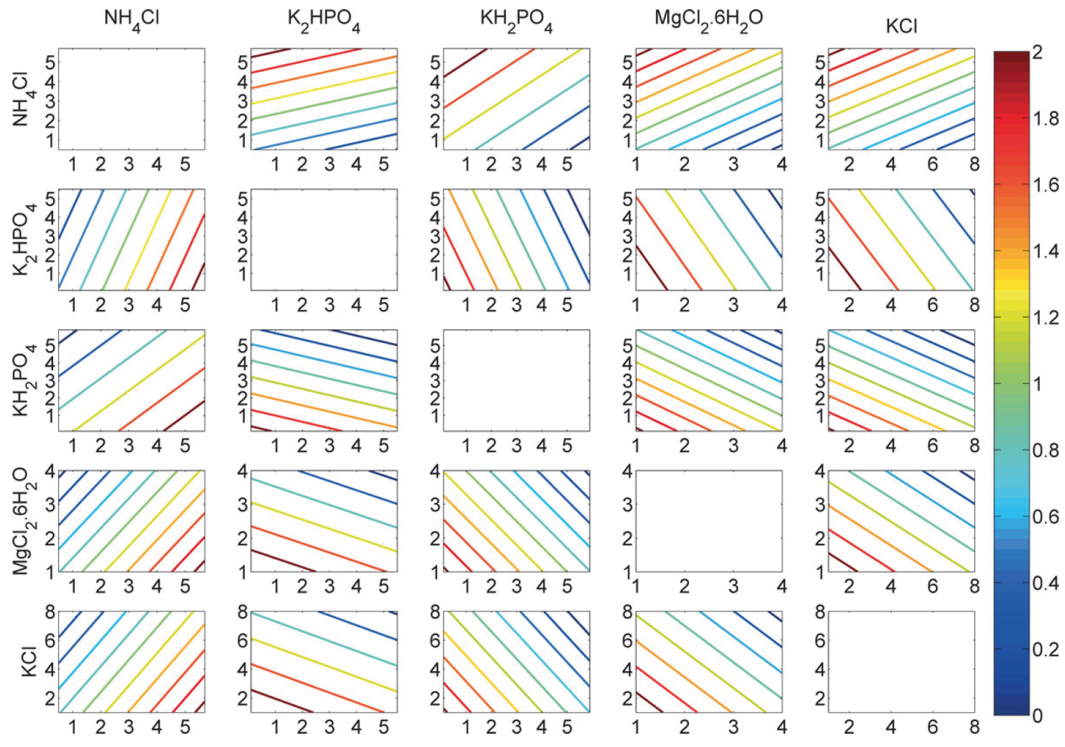


Figure 1 Yield predictions using the regularized regression model. The yields are presented by different colors according to the colorbar. The plots in the diagonal (i.e., variables are plotted against themselves) are left empty.

Table 1 Significant variables and their coefficients in the regularized regression model

Significant variables	Coefficient values
NH_4Cl	0.1254
K_2HPO_4	-0.0383
KH_2PO_4	-0.1061
$MgCl_2.6H_2O$	-0.1418
KCl	-0.0562

were used as additional predictor variables. Eventually, the increase is quite small, and may thus be a due to random fluctuation.

Method comparison

Both regularized regression with transformed variables and random forests produced results that are useful in bioprocess data mining. In particular, both methods determined all the variables significant and can be used to determine an advantageous control direction for them. The most notable difference in the results is the linearity that was in use in the regularized regression versus the nonlinearity that is inherent in random forests (see Figures 3 and 4). Simple linear models cannot fit to the nonlinearity of the data and, thus, the maximum response cannot be detected inside the examined space although it would be located in there. However, regularized linear regression with transformed variables was found successful in modeling the nonlinearity of the data to some extent. On the other hand, the random forest model is able to capture the nonlinearity. Here, the maximum response was determined approximately at the same point as in the media optimization study performed using the methods of statistical design of experiments.

Figure 3 and 4 show the performance of the three methods in yield prediction. It is clear that regularized linear regression failed to cope with data non-linearity unless transformed variables were used in regression. On the other hand, the use of transformed variables causes the multiple linear regression to fail. Thus, multiple linear regression is an efficient tool in the analysis of individual datasets designed by statistical design of experiments (e.g.,

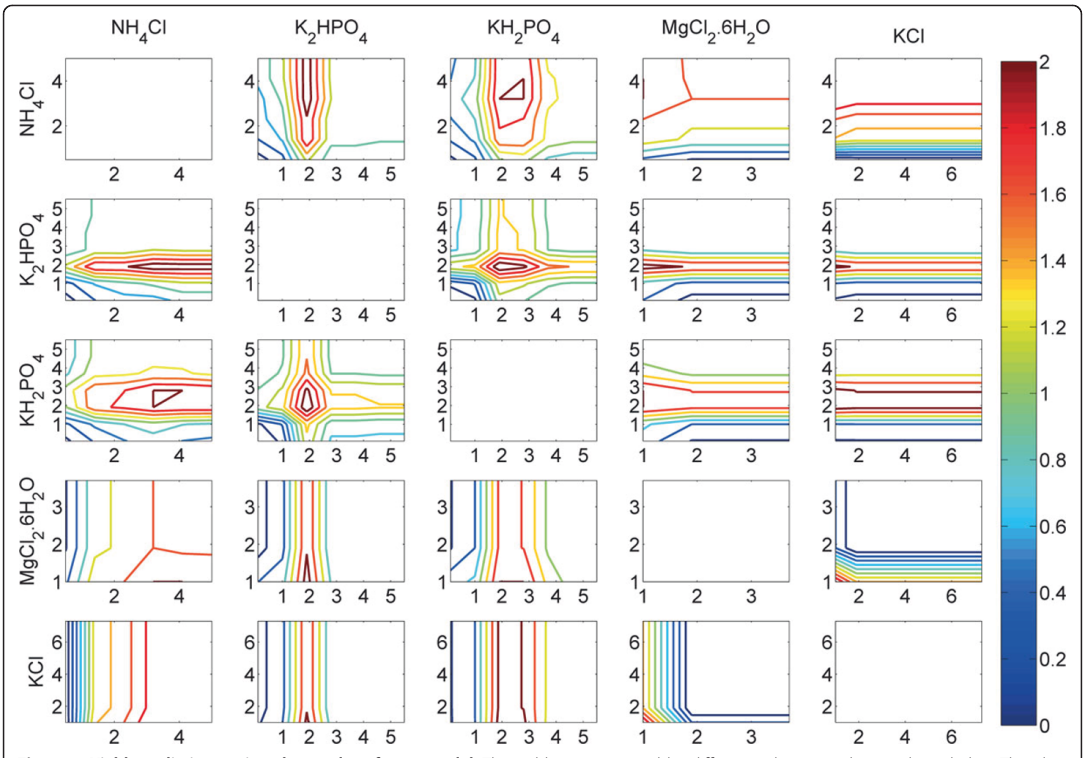


Figure 2 Yield predictions using the random forest model. The yields are presented by different colors according to the colorbar. The plots in the diagonal (i.e., variables are plotted against themselves) are left empty.

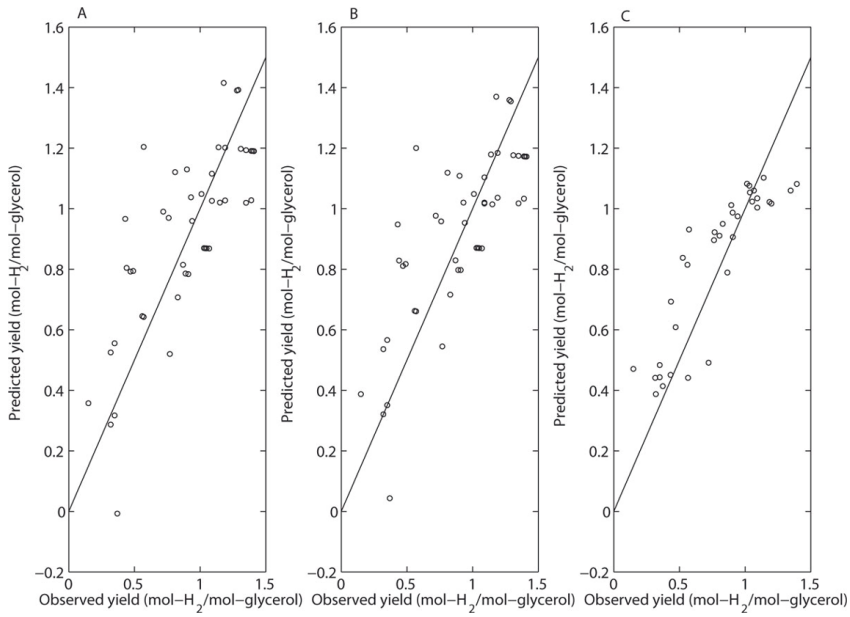


Figure 3 Comparison of prediction performance of models obtained by three methods for original dataset. **(A)** Multiple Linear Regression; **(B)** Lasso; **(C)** Random Forest. The straight line depicts perfect predictions should lie. The prediction accuracy for each model is estimated using LOO cross-validation.

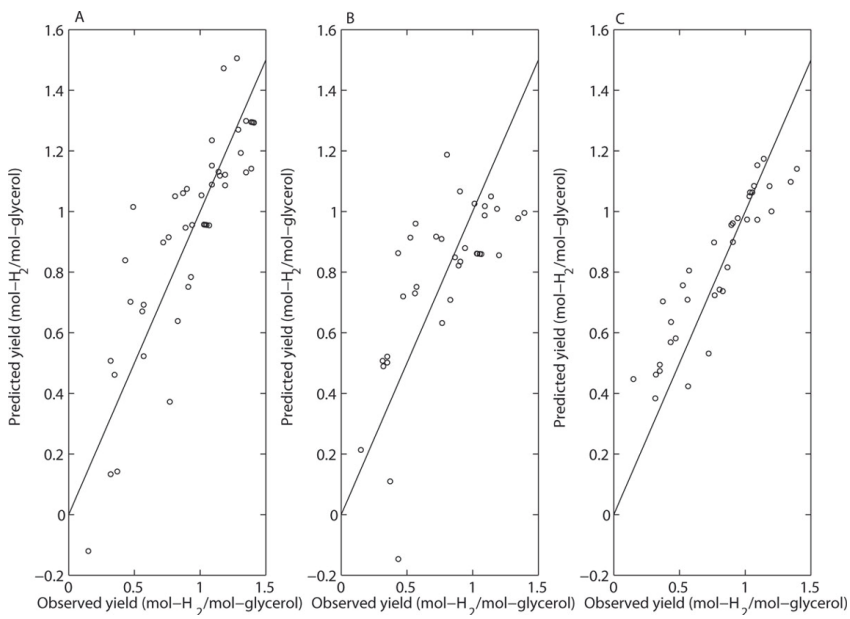


Figure 4 Comparison of prediction performance of models for the dataset containing the actual and the transformed variables. **(A)** Multiple Linear Regression; **(B)** Lasso; **(C)** Random Forest. The straight line depicts perfect predictions should lie. The prediction accuracy for each model is estimated using LOO cross-validation.

Plackett-Burman and Box-Behnken) but not useful in data mining of more complicated datasets like the one examined in here.

The LOO estimates for correlation ascertain that the RF-ACE provides a more accurate solution than the regularized regression. This, however, should not totally renounce the idea of using regularized regression as it mainly proves its worth in more complicated and high-dimensional data analysis. Moreover, linear regression has a useful feature of producing easily interpretable models and, on the other hand, the models are capable in producing predictions beyond the already examined parameter space.

Conclusions

In this study, we applied two novel data analysis methods (regularized regression and random forests) in bioprocess data mining and compared them to multiple linear regression that is commonly applied in relation to statistical design of experiments. Both of the studied methods were able to produce models that fit to the examined data. In particular, the non-linearity of the data was well modeled by random forests. This property is very valuable in data mining of multiple integrated data sets. As the results demonstrated, traditionally used multiple linear regression does not perform satisfactorily in non-linear input-output relations. The traditional approach using the first and the second order polynomial models would face further problems if the data was multimodal. In the future, it would be of interest to further study regularized regression and random forests in bioprocess data mining. This could mean, for example, the inclusion of categorical variables in the data and studies with different types of bioprocesses.

Additional material

Additional file 1: as PDF - Table S1: Significant coefficient values in different methods using transformed data. This file contains a table describing the coefficient values generated by Lasso and multiple linear regression methods for the transformed dataset. Here, the coefficient β_0 represents the intercept, β_1 corresponds to variable NH_4Cl, β_2 to K_2HPO_4, β_3 to KH_2PO_4, β_4 to $MgCl_2.6H_2O$ and β_5 to KCl, respectively.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

SSH and MF made substantial contribution in writing the manuscript, interpretation of the data, and design and analysis of the models. RM was responsible in acquisition of the data. TA and HH contributed to the design of the study, and in writing and revising the manuscript.

Acknowledgements

The authors thank the Academy of Finland, project "Butanol from Sustainable Sources" (decision number 140018), for funding the study.

Declarations

The publication cost for this work was supported by the Academy of Finland, project "Butanol from Sustainable Sources" (decision number 140018).

This article has been published as part of BMC Systems Biology Volume 7 Supplement 1, 2013: Selected articles from the 10th International Workshop on Computational Systems Biology (WCSB) 2013: Systems Biology. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcsystbiol/supplements/7/S1.

Authors' details

[1]Department of Signal Processing, Tampere University of Technology, Tampere, P.O. Box 553, 33101, Finland. [2]Department of Chemistry and Bioengineering, Tampere University of Technology, Tampere, P.O. Box 541, 33101, Finland.

Published: 12 August 2013

References

1. CDER: **Process Validation: General Principles and Practices.** 2011 [http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/ucm070336.pdf].
2. Tuv E, Borisov A, Runger G, Torkkola K: **Feature selection with ensembles, artificial variables, and redundancy elimination.** *Journal of Machine Learning Research* 2009, **10**:1341-1366.
3. Tibshirani R: **Regression shrinkage and selection via the Lasso.** *J R Statist Soc B* 1996, **58(1)**:267-288.
4. Mathworks: *Matlab* Natick, MA; 2011.
5. **RF-ACE: multivariate machine learning with heterogeneous data.** [http://code.google.com/p/rf-ace/].
6. Andersen PK, Skovgaard LT: **Multiple regression, the linear predictor.** In *regression with linear prediction. Volume 0.* New York, NY: Springer; 2010:231-302.
7. Miller AJ: **Subset Selection in Regression.** Chapman and Hall/CRC;, 2 2002.
8. Breiman L: **Random forests.** *Machine Learning* 2001, **45(1)**:5-32.
9. Saeys Y, Inza I, Larranaga P: **A review of feature selection techniques in bioinformatics.** *Bioinformatics* 2007, **23(19)**:2507-2517.

doi:10.1186/1752-0509-7-S1-S5
Cite this article as: Hassan *et al.*: Bioprocess data mining using regularized regression and random forests. *BMC Systems Biology* 2013 **7**(Suppl 1):S5.

Publication II

Sara Urmersbach, Tommi Aho, Thomas Alter, Syeda Sakira Hassan, Reija Autio, and Stephan Huehn. "Changes in global gene expression of Vibrio parahaemolyticus induced by cold- and heat-stress", *BMC microbiology*, vol 15, no. 1, pp. 229, Oct. 2015.

Urmersbach et al. BMC Microbiology (2015) 15:229
DOI 10.1186/s12866-015-0565-7

BMC Microbiology

RESEARCH ARTICLE Open Access

Changes in global gene expression of *Vibrio parahaemolyticus* induced by cold- and heat-stress

Sara Urmersbach[1], Tommi Aho[2], Thomas Alter[1], Syeda Sakira Hassan[2,3], Reija Autio[3,4] and Stephan Huehn[1*]

Abstract

Background: *Vibrio* (*V.*) *parahaemolyticus* causes seafood-borne gastro-intestinal bacterial infections in humans worldwide. It is widely found in marine environments and is isolated frequently from seawater, estuarine waters, sediments and raw or insufficiently cooked seafood. Throughout the food chain, *V. parahaemolyticus* encounters different temperature conditions that might alter metabolism and pathogenicity of the bacterium. In this study, we performed gene expression profiling of *V. parahaemolyticus* RIMD 2210633 after exposure to 4, 15, 20, 37 and 42 °C to describe the cold and heat shock response.

Methods: Gene expression profiles of *V. parahaemolyticus* RIMD 2210633 after exposure to 4, 15, 20, 37 and 42 °C were investigated via microarray. Gene expression values and RT-qPCR experiments were compared by plotting the log2 values. Moreover, volcano plots of microarray data were calculated to visualize the distribution of differentially expressed genes at individual temperatures and to assess hybridization qualities and comparability of data. Finally, enriched terms were searched in annotations as well as functional-related gene categories using the Database for Annotation, Visualization and Integrated Discovery.

Results: Analysis of 37 °C normalised transcriptomics data resulted in differential expression of 19 genes at 20 °C, 193 genes at 4 °C, 625 genes at 42 °C and 638 genes at 15 °C. Thus, the largest number of significantly expressed genes was observed at 15 and 42 °C with 13.3 and 13 %, respectively. Genes of many functional categories were highly regulated even at lower temperatures. Virulence associated genes (*tdh*1, *tdh*2, *tox*R, *tox*S, *vop*C, T6SS-1, T6SS-2) remained mostly unaffected by heat or cold stress.

Conclusion: Along with folding and temperature shock depending systems, an overall temperature-dependent regulation of expression could be shown. Particularly the energy metabolism was affected by changed temperatures. Whole-genome gene expression studies of food related pathogens such as *V. parahaemolyticus* reveal how these pathogens react to stress impacts to predict its behaviour under conditions like storage and transport.

Keywords: *Vibrio parahaemolyticus*, Gene expression, Thermal shock

Background

Vibrio (*V.*) *parahaemolyticus* is one of the causes of seafood-borne gastro-intestinal infections in humans worldwide [1]. It is widely found in marine environments and is isolated frequently from seawater, estuarine waters, sediments and raw or insufficiently cooked seafood (e.g. shrimp or bivalve molluscs) [2–4]. Consumption of or contact to raw or undercooked seafood containing *V. parahaemolyticus* in relevant numbers, might lead to human infections, mostly associated with gastroenteritis [5, 6].

Different studies investigated the behaviour of *V. parahaemolyticus* under environmental stresses on the phenotypic level (e.g. cold shock, heat shock, high salt concentrations or bile supplementation) [7–9]. Nonetheless, the general mechanism of adaptation and survival under these conditions are not elucidated yet.

Within its ecological habitat and food chain, *V. parahaemolyticus* encounters changing temperature conditions. These temperature shifts will result in metabolic changes.

* Correspondence: stephan.huehn@fu-berlin.de
[1]Institute of Food Hygiene, Freie Universität Berlin, Berlin, Germany
Full list of author information is available at the end of the article

A cold shock resulting from a rapid downshift of the temperature, e.g. changing water temperatures or storage on ice, alters bacterial gene expression [10–13]. However the expression of *V. parahaemolyticus* resulting from cold shock is still poorly understood. The cold-induced gene expression profile of a clinical *V. parahaemolyticus* strain at 10 °C has been examined by Yang et al. [13] in a time course analysis. Significant differential expression of almost 13 % of genes (n = 619) investigated, was found.

Temperatures in the marine habitat of *V. parahaemolyticus* usually do not exceed 25 °C. In *V. parahaemolyticus* several stress proteins, e.g. heat shock protein (*hsp*) families such as Hsp60 (GroEL and GroES) and as Hsp70 (DnaJ, DnaK, GrpE) are produced in response to elevated temperatures [14, 15]. In general those proteins are made in substantial amounts acting as chaperones, protecting cells from heat dependent denaturation [16–18]. In *V. parahaemolyticus* especially Hsp60 family proteins serve as general stress proteins and are found in several cell compartments and in substantial amounts [19].

In addition, changing temperature conditions can affect the pathogenicity of *V. parahaemolyticus* [19]. Chiang and Chou [20] demonstrated increased pathogenicity after heat shock response in *V. parahaemolyticus* as elevated toxin expression. Clinical strains alter expression of systems regulating virulence as well as systems indirectly related to host-pathogen attachment such as biofilm production and motility at 37 °C [21]. However, environmental strains did not show this behaviour or exhibit decreased expression of biofilm production or motility related genes at higher temperatures. Sublethal heat shock of *V. parahaemolyticus* resulted in elevated expression levels of the gene encoding the thermostable direct hemolysin (TDH), one of the two prominent toxins enhancing its pathogenicity [19].

The aim of this study was to investigate gene expression profiles of *V. parahaemolyticus* after exposure to 4, 15, 20, 37 and 42 °C. Moreover high regulation clusters e.g. toxins produced in response to temperature changes were to be identified.

Results and discussion

Understanding temperature-dependent changes in bacterial gene expression patterns is crucial when studying tenacity, invasion, and environmental related viability of bacterial species. Temperature-dependent expression changes as cues for tenacity and persistence within matrixes such as food or hosts and environment has led to genetic approaches defining temperature-induced genes of pathogens [22–24]. However, temperature-dependent induction of genes is an arbitrary parameter because appropriate temperatures for comparison to any other temperature must be assumed. In this study, we investigated temperature-dependent gene expression of *V. parahaemolyticus* in comparable growth phases under different temperatures.

Validation of microarray results

To confirm the results of microarray data analysis, a quantitative RT-qPCR was used. Six house-keeping genes were chosen to compare the data of the two techniques, whereof four were applied in the multilocus sequence typing (MLST) scheme of *V. parahaemolyticus* [25]. The house-keeping genes were encoded on both chromosomes, with one exception (*csp*A): *csp*A, *dtd*S, *gro*ES, *pvs*A, *pyr*C and *tna*A. Four additional MLST genes used for normalization: *pvu*A, *dna*E, *rec*A and one locus of the 16S-23S intergenic spacer region. Gene expression values of microarray and RT-qPCR experiments were compared by plotting the $\log_2$ values of both experiments against each other. An overall positive correlation ($R^2 = 0.7008$) between the two techniques could be shown (Fig. 1). The similarity of replicate samples at different temperatures was studied using hierarchical clustering with correlation as the distance measure (Fig. 2a). The samples at 42 °C form the clearest cluster. Samples of 20 and 37 °C cluster according to the temperature. Moreover, volcano plots of microarray data were calculated to visualize the distribution of differentially expressed genes at individual temperatures and to assess quality and comparability of hybridizations (Fig. 2b). Additionally, volcano plots enable the quick identification of expression changes within the gene sets by combination of statistical tests (adjusted p-value) and magnitude of changes.

Gene expression at 4 °C, 15 °C, 20 °C and 42 °C

We compared gene expression patterns within a temperature range of 4 to 42 °C. Additionally, Database for Annotation, Visualization and Integrated Discovery (DAVID) analyses were performed, highlighting regulation of genes connected in metabolic pathways (Additional file 1). Among all conditions, the strongest expression changes (regarding the number of differentially expressed genes and intensity of expression changes) were observed at 15 °C (13.3 % of all genes) and 42 °C (13 % of all genes). Since the highest number of genes with stable expression was found at 37 °C, this temperature was chosen as reference. Genes with an adjusted p-value ≤0.05 and an absolute logarithmic fold change ≤±1.5 were considered significantly stable expressed. To demonstrate the temperature-associated differences in gene expression changes, temperature experiments were clustered via K-means-clustering (Fig. 3). The K-means-clustering arranges genes showing comparable expression under all temperatures investigated. Some genes showing clear up-regulation in both extreme conditions [Fig. 3 - cluster eight, 275 genes including sugar transport

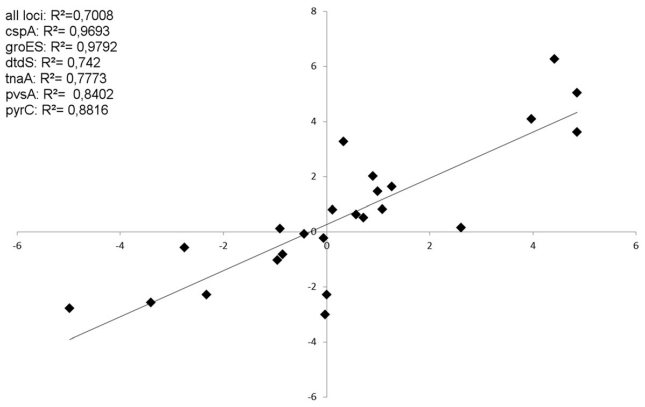


Fig. 1 Correlation of microarray and qRT expression of selected genes and quality control. Log_2 transformed values. R^2: Coefficient of determination

system permease (VP0908), *tna*A (VPA0192) a tryptopanase, the putative translation elongation factor G, *ptf*G (VPA0328), the putative phosphatase VPA0505, a putative membrane protein VPA1583] were found to be highly upregulated (>2.0 log_2). Genes down-regulated at 4 and 42 °C [Fig. 3 - cluster nine, 410 genes including the D-3-phosphoglycerate dehydrogenase VP2593, *eam*A (VP2828) a pore forming protein and glyoxalase I (VP2166)] have been found in high numbers. In addition, there are genes down-regulated across all temperatures (Fig. 3 - cluster three, 154 genes including the putative proteases VP2447, VP2448 and an alcohol dehydrogenase VPA0870). Expression of genes sorted by chromosomes resulted in a higher rate of differentially expressed genes on the small chromosome (chromosome 2) at 15 and 42 °C (Table 1).

Our analysis identified differentially expressed genes under different temperature conditions. Compared to 37 at 4 °C 4 % ($n = 193$) the genes showed significant expression changes, whereas incubation at 20 °C resulted in a rate of approx. 0.4 % ($n = 19$). At 42 °C, 13 % ($n = 625$) of differentially expressed genes were detected. The highest number of genes regulated, however, was found at 15 °C with 13.3 % ($n = 638$) differentially expressed genes. Incubation at 15 and 42 °C resulted in almost balanced expression patterns regarding the amount of up- and down-regulated genes. At 4 °C, 78 % ($n = 150$) of the significantly differentially expressed genes showed down-regulation, whereas only 22 % ($n = 43$) showed up-regulation. Additional information can be found in Additional file 2.

Expression of temperature shock response genes

Some gene clusters showed up-regulation of expression under one temperature and down-regulation under another. Chaperone encoding *hsp*70 family genes, such as *dna*K (VP0653), as well as the *hsp*60 family *gro*EL, *gro*ES (VPA0286, VPA0287) showed significant down-regulation of expression at 4, 15 and 20 °C. On the contrary, a strong up-regulation at 42 °C was observed (Additional file 2). Cold shock responding genes, such as *cps*A (VPA1289-1291 and VP1889) as well as a cluster encoding genes classified as ascorbate and phosphotransferase (VPA0229-231) showed significant up-regulation at 4, 15 and 20 °C whereas down-regulation occurred at 42 °C. Yang et al. [13] investigated time dependent behaviour of a clinical *V. parahaemolyticus* strain at cold temperatures. Almost 13 % of genes (619 genes) were differentially expressed at least at one of the three points in time investigated [13]. For metabolism related gene categories down-regulation was dominant over up-regulation due to the generally reduced cellular protein pool resulting from a sudden temperature downshift [11].

These findings are confirmed by our data. Moreover under cold temperatures, non-metabolic functions (cell envelope, transport and binding proteins, regulatory functions, cellular processes and mobile and extra-chromosomal element functions) as well as genes with unknown or unassigned functions showed a more frequently up-regulation than genes related to cell structure and trans-membrane transporting functions (Additional file 1). The cold shock protein/regulator CspA (VPA1289) showed an over 30-fold enhanced transcription. Additionally, an antagonistic regulation of cold and heat shock genes was detected: heat shock genes encoding heat shock proteins (*hsp*), ATP-dependent proteases and chaperons were mainly down-regulated after exposure to 10 °C [11–13]. Our results confirm the findings that metabolism related genes at low temperatures were mainly down-regulated and genes without relation to metabolism or of unknown function were mainly up-regulated. Additionally,

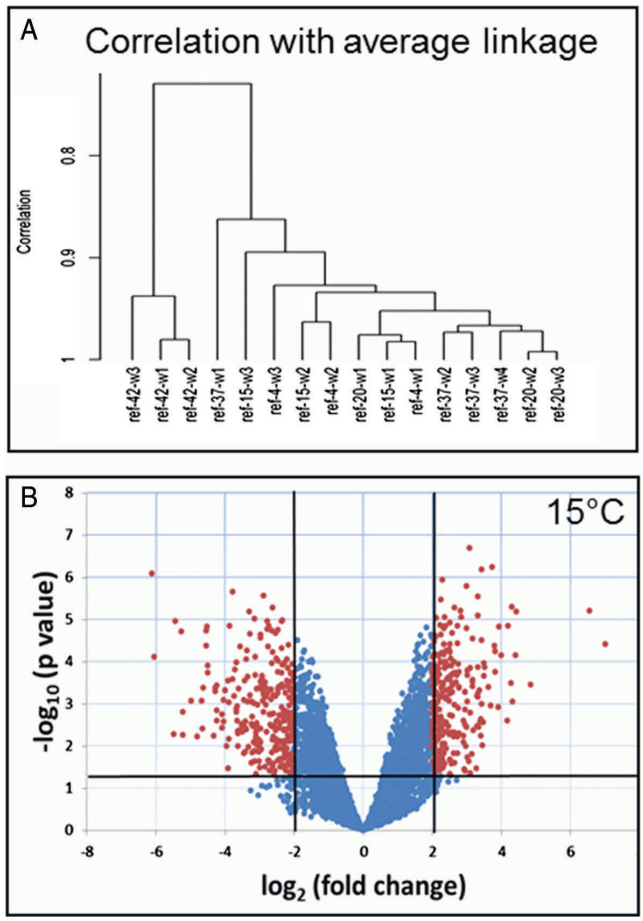


Fig. 2 Overview of microarray results. **a** The dendrogram represents the result of hierarchical clustering with euclidean distance measure. The first number in the sample label represents temperature, the second number is the replicate number at given temperature. **b** Volcano plot exemplarily shown for 15 °C data. The x-axis represents the $\log_2$ of the fold change plotted against the $-\log_{10}$ of the adjusted *p*-value. *Red points* indicate the differentially expressed genes with at least 2.0 fold change and statistical significance adjusted p <0.05

antagonistic expression of cold and heat shock genes as well as a strong induction of cold shock proteins at low temperatures was observed for *V. parahaemolyticus* RIMD 2210633 (Additional file 2).

Gene expression at 4 °C and 15 °C

At 4 °C, only 4 % of the genes were differentially expressed. Primarily transcription regulators as well as RNA metabolic process clusters were up-regulated, highlighting the impact of low temperatures (4 °C) on the overall gene expression (Fig. 4). Phadtare et al. [12] described concordant findings in *E. coli*. In our study, at 4 °C mainly genes encoding hypothetical proteins, e.g. VP1888, VP2889, VP3030 and VPA1291 were up-regulated. Additionally, genes of the energy metabolism (VP1381, VP1533, VP2005, VP2666, VP2987, VPA0092) reacted to 4 °C by up-regulation. Especially VP1533, encoding a putative ATPase, is of great importance for energy production using glucose [26]. In particular cold shock proteins were highly expressed (*csp*A VP1889, 4.05 $\log_2$ fold change). These findings, originally described by Yang et al. [13], were confirmed by our data. However, cold temperatures bias gene expression results due to lower activities of e.g. enzymes [27].

The global regulator sigma factor 38, *rpo*S (VP2553) and the osmoregulator *omp*R (VP0154) were up-regulated (3.5 and 4.1×). Sigma factor 38 is one of the most crucial sigma factors under e.g. extreme temperatures [28]. No

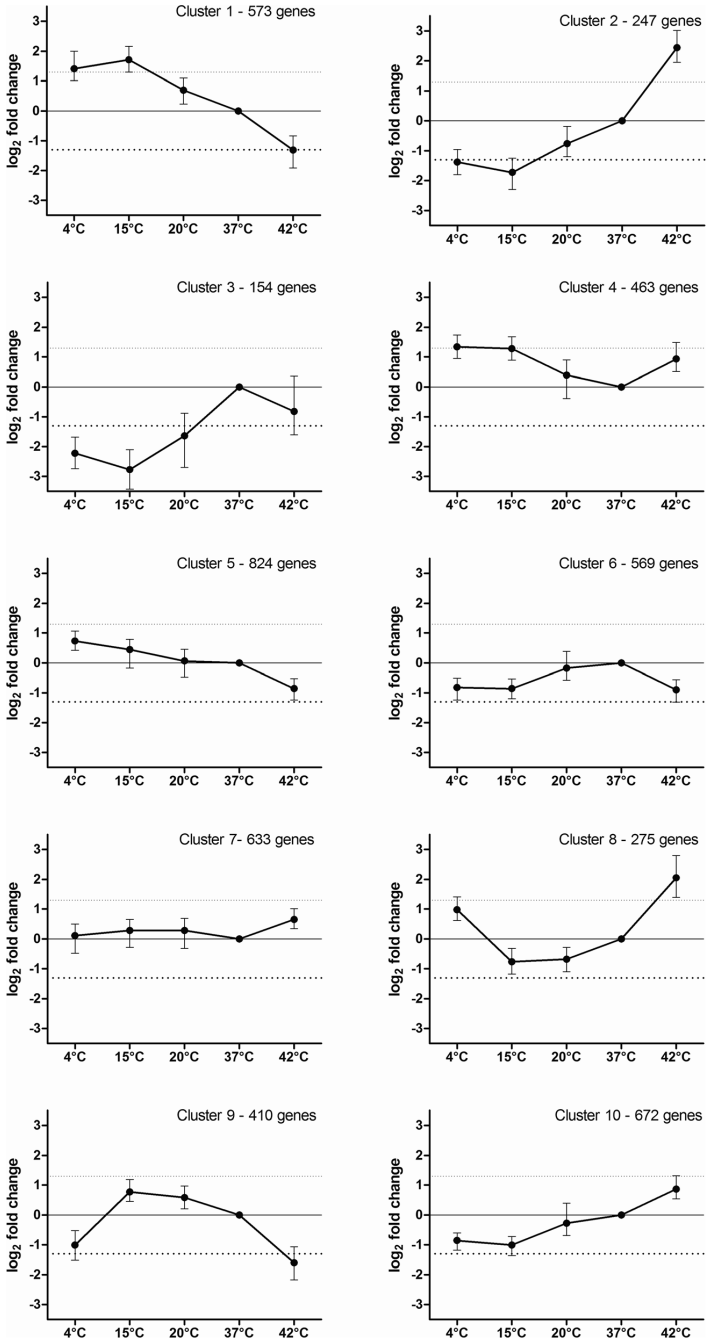


Fig. 3 Clustering of genes with similar expression patterns. Ten clusters of similar expressed genes at 4, 15, 20 and 42 °C normalized to 37 °C are shown. The incubation temperatures (x-axis) where plotted against the x-fold gene expression (y-axis) of genes sorted in the particular box. Clustering was performed using *K*-means (genesis)

Table 1 Differentially expressed genes according to encoding chromosome

Incubation temperature	Chr1-up	Chr1-down	Chr2-up	Chr2-down
4 °C	116 (3.77 %)	26 (0.85 %)	35 (2.00 %)	16 (0.92 %)
15 °C	214 (6.96 %)	171 (5.56 %)	147 (8.41 %)	107 (6.12 %)
20 °C	3 (0.10 %)	9 (0.29 %)	2 (0.11 %)	5 (0.29 %)
42 °C	146 (4.75 %)	204 (6.64 %)	187 (10.70 %)	88 (5.04 %)

The numbers of differentially expressed genes are given in total as well as in proportion to the number of genes present on the microarray in brackets. Chr1: genes encoded on chromosome 1 encoding 3080 genes of which 3073 genes were represented on the array. Chr2: genes encoded on chromosome 2 encoding 1752 genes of which 1747 genes were represented on the array. Down: down-regulated genes; up: up-regulated genes

other sigma factors were up-regulated. Additionally, the tRNA methyltransferase *spo*U (VP0158) described by Persson et al. [29] was up-regulated (4.1×) as well. Persson et al. [29] were not able to detect differences in growth rates of the *E. coli* wild-type and a *spo*U mutant. However, growth temperatures were between 37 and 42 °C in that study. Maybe a particular part of tRNA activation can be triggered by low temperatures. Since none of the known genes related to DNA damage VP2034 (*imu*A), VP2035 (*imu*B), VP2036 (*dna*E2), VP2550 (*rec*A) and VP2945 (*lex*A) were up-regulated, cold induced DNA damaging, triggering the SOS response, appears to be absent. At 4 °C 11 DAVID-gene categories were identified in which a statistically significant number of genes ($n = 186$, p-value <0.05) was differentially regulated. Nine of these categories were related to transcription, DNA-binding and regulation of RNA metabolism. The two other categories were related to ABC-transporters or transmembrane domains. The expression of genes organized into functional categories at 4 °C is shown in Fig. 5. The top five up- and down-regulated genes at 4 °C are shown in Table 2.

At 15 °C a total of 638 genes were differentially expressed (Additional file 2). In addition to *rpo*S, *omp*R and *spo*U, transcriptional regulators VP0034, VP0059, VP0713, VP1391, VP1676, VP1765, VPA0214, VPA1219 and VPA1289 were up-regulated, along with DNA repair VP2943, VPA1393, DNA polymerase III (VP2036), DNA integrase (VP1071). Thus, partially DNA repair has been up-regulated along with protein and peptide secretion and trafficking (VPA1208, VPA1209, VPA1443, VPA 1445). However, no genes related to SOS repair, such as *rec*A (VP2550) and *lex*A (VP2945), or global stress regulators such as *hfq* were up-regulated. These regulators seem of minor concern under these circumstances described. However, strong up-regulation was found for putative regulators such as VP1391 (5×) and VPA1219 (8×).

Especially energy metabolism was down-regulated; out of 75 differentially expressed genes of this category 65

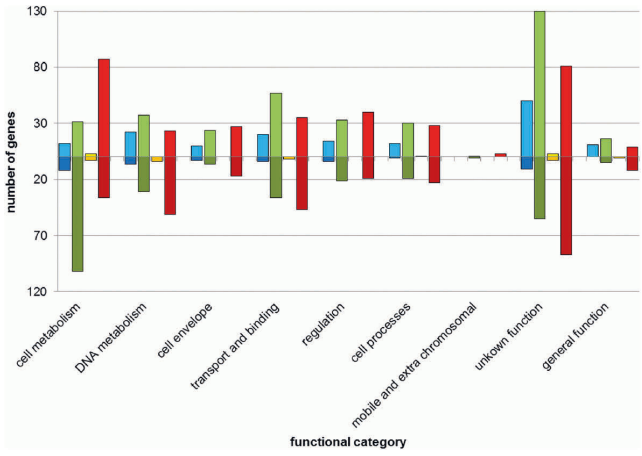


Fig. 4 Functional annotation of differentially expressed genes. The amount of genes was plotted according to their function and incubation temperature. Only significantly expressed genes are shown with at least 1.5 $\log_2$ fold change of expression rate. Normalized incubation temperatures are shown in *blue*, 4 °C; *green*, 15 °C; *yellow*, 20 °C and *red*, 42 °C. *Dark shading* of colour indicates down regulation at the corresponding temperature. All differentially expressed genes with a $\log_2$ fold change >1.5 and adjusted p-value <0.05 in each condition are supplied in Additional file 2

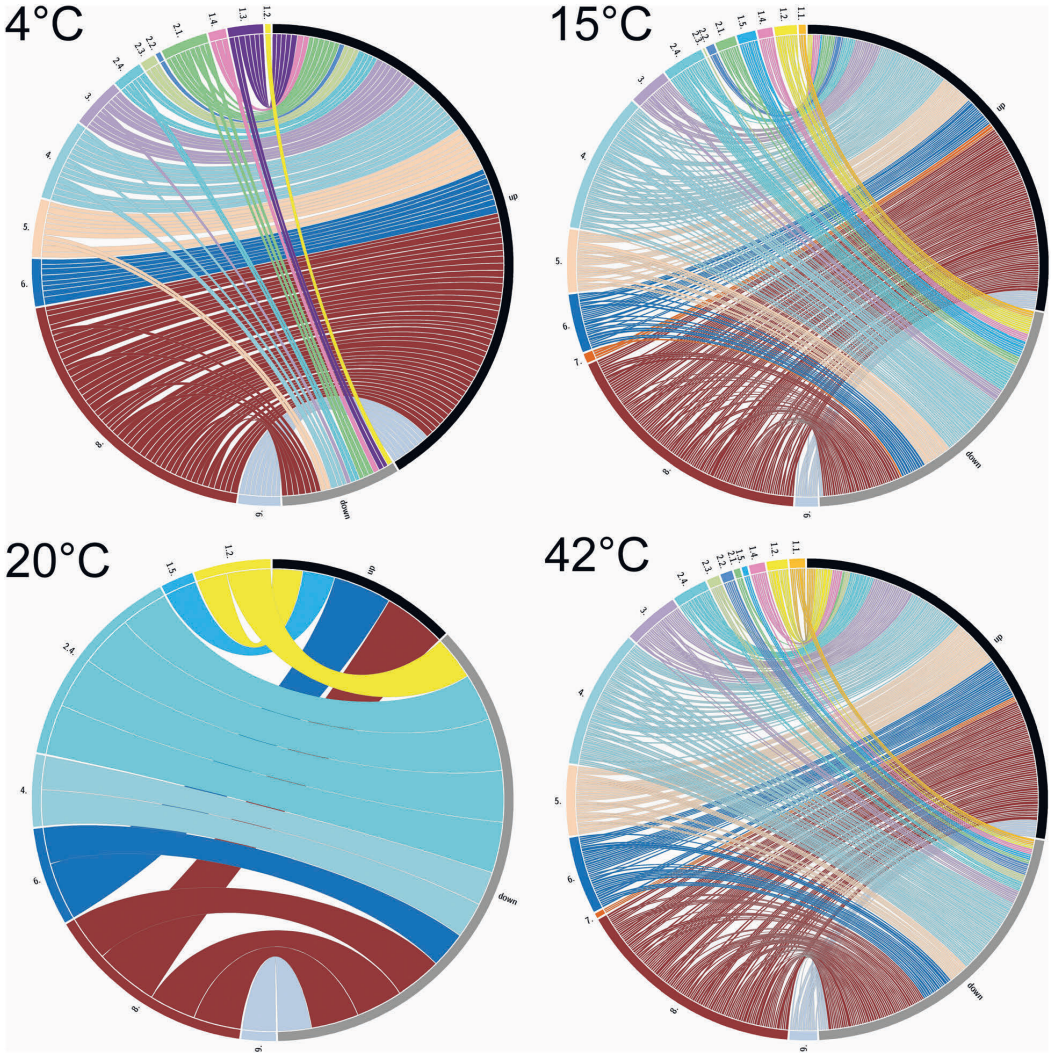


Fig. 5 Integrated graphical view of significantly expressed genes of *V. parahaemolyticus* at 4, 15, 20 and 42 °C. The connections show the direction of regulation in each of the functional groups. *Numbers* indicate functional gene groups (*1.1.* Amino acid biosynthesis, *1.2.* Central intermediary metabolism, *1.3.* Energy metabolism, *1.4.* Fatty acid and phospholipid metabolism, *1.5.* Purines, pyrimidines, nucleosides, and nucleotides, *2.1.* DNA metabolism, *2.2.* Transcription, *2.3.* Protein synthesis, *2.4.* Protein fate, *3.* Cell envelope, *4.* Transport and binding proteins, *5.* Regulatory functions, *6.* Cellular processes, *7.* Mobile and extra-chromosomal element functions, *8.* Unknown, *9.* General function). Gene groups were colored according to functional subgroup. The *black bar* highlights up-regulated genes; the *grey bar* indicates down-regulation

genes (87 %) were repressed (Fig. 4). Genes related to the pentose phosphate pathway, glycolysis and the citric acid cycle were down regulated. We suggest, that production of central energy molecules such as ATP, NADPH and NADH was decreased because of down-regulated expression of corresponding genes. The vast majority of genes showing highest up-regulation, however, were of unknown function (Additional file 2). The putative virulence-associated protein VacB showed highest up-regulation (128×), which has been described to react to environmental signals in *Haemophilus influenza* [30].

Gene expression of functional categories is shown in Fig. 5. In contrast to 4 °C incubation, genes of the amino acid category and *de novo* DNA synthesis were induced

Table 2 Top 5 up- and down-regulated genes at 4 °C

Gene ID	Gene	Protein	log_2 fc 4 °C	adj. p-value
VP1717		ABC-type multidrug	-3.66	0.024
VP2024		hypothetical protein	-3.27	0.017
VP0502		hypothetical protein	-2.83	0.035
VP2264		hypothetical protein	-2.80	0.035
VP2445		put. lipid carrier protein	-2.78	0.018
VP1889	*csp*A	cold shock transcr. reg.	4.06	0.022
VP2889		hypothetical protein	4.15	0.019
VP3030		hypothetical protein	4.46	0.009
VPA1291		hypothetical protein	4.74	0.029
VP1888		hypothetical protein	4.86	0.013

Coloured boxes highlight at least 1.5 fold differential expression in either direction: red up-regulated, green down-regulated, fc (fold change) is given log_2 transformed; *transcr. reg.* transcriptional regulator, *put.* putative

at 15 and 42 °C. In total 32 DAVID-gene categories with 475 differentially expressed genes were identified at 15 °C. Ten categories were associated with transport and transporters. Two major groups formed the most important clusters: integral and intrinsic components of the membrane with 69 (11.6 %) genes each. Additionally, nine metabolism related categories were identified.

At 15 °C 32 DAVID-gene categories with 475 genes showed differential regulation. Many of them were connected with membrane maintenance or metabolism. In *E. coli* it could be shown that at 12 °C the membrane composition remains unchanged but enzyme activations are effected [31]. The top five up- and down-regulated genes at 15 °C are shown in Table 3.

Gene expression at 20 °C

At 20 °C, no differential expression of metabolic pathways was detectable. Thus, a range of temperature between 15 and 20 °C seems to describe the (lower) physiological border of the normal condition for the strain investigated. A total of 19 genes was differentially expressed. Gene expression is shown in Fig. 5. At 20 °C solely genes related to categories associated with the degradation of peptides and proteins were identified: 'peptidase' (15.8 %), 'protease', 'peptidase activity' and 'proteolysis' (21.1 %), respectively. The top five up- and down-regulated genes at 20 °C are shown in Table 4.

Gene expression at 42 °C

At 42 °C transport and metabolism of carbohydrates-related genes were up-regulated (Additional file 2). Again, a range of temperature between 42 and 37 °C seems to describe the (upper) physiological border for the clinical strain investigated. Interestingly, more genes located on the small chromosome were differentially expressed during incubation at all temperatures. Especially, at 42 °C almost twice as many small chromosome genes were differentially expressed. The higher intensity of expression

Table 3 Top 5 up- and down-regulated genes at 15 °C

Gene ID	Gene	Protein	log_2 fc 15 °C	adj. p-value
VP2362	*omp*K	OmpK precursor	-6.10	0.002
VPA0230		put. PTS component II B	-6.05	0.007
VPA1084	*rbs*B	D-ribose transporter	-5.48	0.049
VPA0231		Phosphotransferase	-5.42	0.004
VPA0229	*ula*A	ascorbate-specific PTS	-5.25	0.005
VP1889	*csp*A	cold shock transcr. reg.	4.41	0.007
VPA1413		hypothetical protein	4.44	0.004
VPA1289	*csp*A	cold shock transcr. reg.	4.86	0.014
VP1888		hypothetical protein	6.55	0.004
VP1890	*vac*B	VacB/RNase R	7.01	0.006

Coloured boxes highlight at least 1.5 fold differential expression in either direction: red up-regulated, green down-regulated, fc (fold change) is given log_2 transformed; *PTS* phosphotransferase system; *transcr. reg.* transcriptional regulator, *put.* putative

Table 4 Top 5 up- and down-regulated genes at 20 °C

Gene ID	Gene	Protein	log_2 fc 20 °C	adj. p-value
VPA0611	*ack*A	acetate kinase	-4.60	0.031
VPA1193	*pep*T	peptidase T	-3.92	0.036
VP2448		put. protease	-3.85	0.010
VP2447		put. protease	-3.81	0.010
VP0990		hypothetical protein	-3.56	0.010
VPA0071		put. alcohol dehydrogenase	2.08	0.042
VP1721		aminotransferase	2.25	0.042
VPA1159	*gua*C	guanosine oxidoreductase	2.39	0.018
VP1889	*csp*A	cold shock transcr. reg.	4.38	0.042
VP1888		hypothetical protein	4.75	0.031

Coloured boxes highlight at least 1.5 fold differential expression in either direction: red up-regulated, green down-regulated, fc (fold change) is given log_2 transformed; *transcr. reg.* transcriptional regulator, *put.* putative

changes in genes located on the small chromosome compared to genes located on the large chromosome can be explained by the higher number of genes related to transcriptional regulation and transport of various substances being located on the small chromosome [32]. Thus, most genes related to environmental stress response are encoded on the small chromosome.

Primarily, genes classified as 'cell metabolism' along with the genes classified as 'unknown', reacted to the temperature upshift. Altogether, the expression of 625 genes was differentially expressed at 42 °C. Expression of categorized genes is shown in Fig. 5. Out of the 87 'cell metabolism' genes, 55 % ($n = 48$) were classified as 'energy metabolism' related genes.

A wide spectrum of genes was affected. For example, genes associated with amino acid and amine synthesis (pyruvate family) were induced, whereas genes related to histidine (VP1137), serine (VP1324, VP1629, VP2593) and aromatic amino acid (VP2744, VP3065) families were down-regulated (Additional file 1). Out of 55 energy metabolism related genes only six were down-regulated in expression. Particularly, genes of electron transfer (VP1161, VPA0643, VPA0949, VPA1428), biosynthesis of polyamines (VPA0169, VPA0170, VPA1635) and degradation of fatty acids as well as fermentation (VP1647, VP2543, VPA0478, VPA0502, VPA1416) were up-regulated at 42 °C. Moreover especially sugar metabolism (VP1303, VP2397, VP2398, VP2400, VPA1674, VPA1675, VPA1700, VPA1706) was affected (Additional file 1). Genes involved in arabinose (VPA 1671–1678), mannose and glucoronate (VPA1702-1709) metabolism and transport were up-regulated. Additionally heat protection protein encoding genes such as *gro*EL, *gro*ES were induced. Reactions of heat shock proteins such as GroEL/GroES, are in concordance with data described by Wong et al. [19].

At 42 °C, 38 DAVID-gene categories with a total of 423 differentially expressed genes were identified (Additional file 1). Amongst others, nine categories were related to cell-motion (flagella), eight categories to metabolic processes and six categories to RNA, DNA and transcription. Additionally, three categories were associated with homeostasis (ion, cation, chemical) and two categories with iron-siderophores and transport of siderophores. A distinct cluster on the second chromosome encoding the genes VPA0915-1042 ('cellular processes': $n = 23$, 'energy metabolism': $n = 18$, 'transport and binding': $n = 14$, 'regulatory functions': $n = 14$ and 'unknown': $n = 36$) showed up-regulation at 42 °C. The top five up- and down-regulated genes at 42 °C are shown in Table 5.

However, no prior studies about genome wide gene expression responses exist for the temperatures investigated in this study.

Temperature dependent expression of virulence genes

Virulence genes in total showed no significant expression changes under different temperatures (Additional file 2). The expression of *tdh* was not significantly influenced by temperature changes, even though slight activation (2.1 log_2 fold change) was observed at 15 °C. A putative haemolysin encoding gene (VP3048), was up-regulated at 4 and 15 °C. This effect was described by Yang et al. [13], reporting an induction of this putative haemolysin after cold shock. The most prominent haemolysin *tdh*, however, was not significantly up-regulated (Additional file 2). The associated regulator *opa*R which recently has been shown to repress expression of T6SS in *V. parahaemolyticus* is down-regulated at 42 °C [33]. We found that, genes located within the virulence pathogenicity island 7 (VPa-7) encoded on the small chromosome, VPA1312-1396 showed no reaction to thermal stimulations (Additional file 2). However, since the energy metabolism was affected

Table 5 Top 5 up- and down-regulated genes at 42 °C

Gene ID	Gene	Protein	log_2 fc 42 °C	adj. p-value
VP0712		hypothetical protein	-5.64	8.16546E-07
VPA1424		PTS system	-5.42	0.016
VP0053		hypothetical protein	-5.10	0.011
VPA1289	*csp*A	cold shock transcr. reg.	-4.98	0.022
VPA1425	*man*A	mannose-6-p isomerase	-4.89	0.010
VP0084		hypothetical protein	5.57	0.026
VP2479		peptide ABC transporter	5.57	0.002
VPA0505		put. hydrolase	6.13	0.010
VPA0286	*gro*ES	co-chaperonin GroES	6.24	0.005
VPA0287	*gro*EL	chaperonin GroEL	6.66	0.004

Coloured boxes highlight at least 1.5 fold differential expression in either direction: red up-regulated, green down-regulated, fc (fold change) is given log_2 transformed; *transcr. reg.* transcriptional regulator, *put.* putative

especially and mostly at cold temperatures, reduced classical virulence or changed expression rates were to be expected [34].

Virulence associated genes in general (*tdh*1, *tdh*2, *tox*R, *tox*S, *vop*C, T6SS-1: VP1386-1420), remained unaffected by heat or cold stress (Additional file 2). The T3SS-1 was found down-regulated at 15 °C for *nos*A (VP1697) and up-regulated for the putative chaperone VP1687 at 42 °C. However, the T6SS-1 located on chromosome 1 showed up-regulation at 42 °C. This was to be expected since the T6SS-1 system reacts to warm climate in *V. parahaemolyticus* as described by Salomon et al. [35]. The cold shock gene *csp*A was down-regulated, whereas heat shock genes encoding chaperones and protection via sugar metabolites were induced [13].

Conclusions

Based on our data, the optimal temperature range of the clinical *V. parahaemolyticus* strain investigated is between 20 and 37 °C, since most of the genes were transcribed at a rather constant level.

Finally, it could be shown that the classical pathogenicity markers, T3SSs as well as T6SSs were not up-regulated in response to thermal changes. However, large proportions (~30 %) of the differentially expressed genes are of unknown function. Summarized, this study successfully demonstrated that genome-wide gene expression changes in *V. parahaemolyticus* occur at 4, 15, 20, and 42 °C.

Methods

Bacterial strains

V. parahaemolyticus RIMD2210633 was isolated from a patient suffering from diarrhoea in Japan in 1996 [32]. This strain harbours the *tdh* gene, lacks the *trh* gene and belongs to serotype O3:K6 [36]. This serotype has been detected in clinical as well as in environmental marine samples [37]. The strain has been sequenced by Makino et al. [32].

Prior use, the strain was stored in cryovials at −80 °C (Cryobank; Mast Diagnostica, Bootle, England). For initial growth, cells were grown using a rotary shaker (Unimax 1010 and Incubator 1000; Heidolph, Schwabach, Germany) in alkaline peptone water (APW; 0.3 % Yeast-Extract, 1 % Peptone, 2 % NaCl; pH 8.6) at 37 °C overnight. A 2 ml aliquot of the resulting culture was diluted to a total volume of 25 ml using APW and grown to an $A_{600\ nm}$ of 0.6. Cultures were grown at 37 °C for 3.5 h in order to generate exponential phase cultures. After appropriate dilutions the A_{600} was analysed again and aliquots consisting of 10^8 to 10^9 *V. parahaemolyticus* cells were transferred to 15 ml Falcon tubes, placed in a thermal mixer (Thermomixer comfort; Eppendorf, Hamburg, Germany) and incubated at different temperatures (42, 37 and 20 °C) for 30 min. For stressing the cells at 4 and 15 °C the entire incubation unit was placed in a conditioning cabinet (Rubarth Apparate, Laatzen, Germany) and bacteria were incubated at these temperatures for 30 min.

RNA preparation and reverse transcription for qPCR investigation

The cultures were centrifuged (2 min, 8000 × g) and the supernatant was discarded. The pellet was immediately resuspended in 1.5 ml RNAprotect Bacteria Reagent (Qiagen, Hilden, Germany) to minimize RNA degradation. Total RNA was isolated using the peqGold Bacterial RNA Kit (Peqlab, Erlangen, Germany). The obtained RNA was eluted into 43 μl of DEPC-treated, DNase- and RNase-free water (Carl Roth, Karlsruhe, Germany). Samples were then treated with DNase I along with Ribolock, an RNase-A, −B and -C inhibitor (Fermentas, Vilnius, Lithuania). RNA quantity was measured by spectrophotometry. RNA quality of each sample was

monitored via gel electrophoresis. Additionally, the RNA quality was assessed using the Agilent RNA 6000 Nano Kit on a 2100 Bioanalyzer (Agilent, Santa Clara, US).

Fluorescence-labeled cRNA generation for the microarray

Prior to labelling, the RNA was initially transcribed in cDNA. Briefly, 200 ng of RNA were linear amplified using the full spectrum MultiStart primer (Biocat, Heidelberg, Germany) and Moloney murine leukemia virus reverse transcriptase (Agilent). The amplification was performed at 40 °C for 2 h followed by 65 °C for 15 min and stored at 4 °C. The amplified cDNA, the full spectrum MultiStart primer and T7 RNA polymerase were used along with Cyanine 3-CTP (Agilent) generating labelled cRNA. Labeling was performed using the Quick Amp Labeling Kit (Agilent). The labeled cRNA was purified using the Qiagen RNeasy Mini Kit (Qiagen). A 3 μl-aliquot was used for quality control. Experiments were performed using Agilent custom 8 × 15 k arrays (Agilent). The microarray field covers 99.75 % of all *V. parahaemolyticus* genes. In total, 3073 out of 3080 genes encoded on chromosome 1 and 1747 out of 1752 genes located on chromosome 2, are included. Each gene is represented by 1 to 10 probes (mean 3.15 probes per gene). Each probe consists of a 60mer located preferentially at the 3' terminus of the corresponding gene. The probe design was performed with the eArray Software a web-based Agilent application basing on the genome sequence of *V. parahaemolyticus* RIMD 2210633 (http://www.ncbi.nlm.nih.gov/genome/691?genome_assembly_id=167995). The cRNA samples were then hybridized to an individual microarray field.

Microarray hybridization and post hybridisation washing

For hybridizations on the microarray, three replicates of independently grown bacterial cultures were prepared for each temperature condition, for 37 °C four replicates were used. Accordingly, three individually labeled cRNA sets were prepared for each temperature other than 37 °C. Finally, 600 ng of the labeled and linear amplified cRNA was fragmented, added to 25 μl hybridization buffer mix of which a 20 μl aliquot (480 ng) was loaded on a microarray in a hybridization chamber (Biometra, Goettingen, Germany). The one-channel hybridization was performed at 65 °C for 17 h and 10 rpm.

Washing of the slides was performed using preheated washing buffer (Gene expression wash buffer kit, Agilent). First the chamber was rinsed with washing buffer. Then the slides were washed once followed by a second washing step using washing buffer containing 0.01 % Triton X-102 (Agilent). The slides were dried using acetonitrile.

Table 6 qRT-PCR primers

Gene	ID	Sequence 5′ to 3′	Size [bp]	Reference
1623S	bp 134385	GCTGACAAAACAACAATTTATTGTT	170	[45]
	to 135166[a]	GGAGTTTCGAGTTGATGAAC		
*gro*ES	VP2852	TATTCAACGATCGCCATGAT	108	This study
		TGGTGACACCGTTATCTTCG		
*csp*A	VPA1289	TATCGTTGCTGACGGTTTCA	90	This study
		TCAGTCGCTTGAGGACCTTT		
*pvs*A	VPA1658	GGACCTCCACGTCGTTCTTA	112	This study
		GGGATTGAAGACATCGCACT		
*pvu*A	VPA1656	GCTGTCGATGCTTGATCGTA	107	This study
		GTGGAATCGGTTTGGTCACT		
*rec*A	VP2550	GAAACCATTTCAACGGGTTC	139	[25]
		GTGCAGCAGCGATAAGCTC		This study
*dna*E	VP2303	GATTACCGCTTTCGCCG	140	[25]
		GTGTATCCATGCCCGATTTC		This study
*dtd*S	VPA1508	TGGCCATAACGACATTCTGA	124	[25]
		TTCGTGACCGACAACCATAG		This study
*pyr*C	VPA0408	AGCAACCGGTAAAATTGTCG	142	[25]
		TCCATGAACCAAAAGCAACA		This study
*tna*A	VPA0192	TGTACGAAATTGCCACCAAA	103	[25]
		TCAGCGTAACCTTCTTCACG		This study

[a] 16S-23S intergenic spacer region encoded on chromosome 2

Data handling and microarray analysis

Scanning was carried out using the Agilent G2565CA scanner with a resolution of 5 μm. After scanning, tiff-files were analysed and raw data was extracted using Feature Extraction Software (Agilent). Data processing was performed using Bioconductor V 2.12 package of the software R. At first, background corrected spot intensities (signal gProcessedSignal in the Agilent protocol GE1_107_Sep09) were retrieved and bad quality spots were removed using the outlier detection flags of the Agilent protocol. Further, the signal values were normalized using quantile normalization and $\log_2$ transformed [38]. Linear modelling and empirical Bayes methods, implemented in the R package Limma [39], were used to detect the differentially expressed genes between two groups, in this case, the control and treatment sample groups. Raw *p*-values were adjusted using the Benjamini and Hochberg multiple adjustment method [40]. Genes with an adjusted *p*-value ≤0.05 and an absolute logarithmic fold change ≥1.5 were considered significantly differentially induced, while genes with an absolute logarithmic fold change ≤ −1.5 were considered repressed. Annotation of genes was performed according to Yang et al. [13] and updated using two new gene entries at NCBI (http://www.ncbi.nlm.nih.gov/gene), KEGG (http://www.genome.jp/kegg/) and Gene Ontology (http://www.geneontology.org/).

Finally, enriched terms were searched in annotations as well as functional-related gene categories using the Database for Annotation, Visualization and Integrated Discovery (DAVID V 6.7, Fisher exact test) [41, 42]. The gene lists generated via DAVID enable to highlight gene sets which show a higher proportion of differentially expressed genes compared to other categories. This eases identification of pertinent biological processes to the according temperature. The identified categories are presented in the Additional file 1. *K*-means clustering of genes with similar gene expression was performed using Genesis V 1.7.6 [43]. Heat maps were generated using BioNumerics V 6.01 (Applied Math, St. Martens-Latem, Belgium). Volcano plots were generated via GraphPad V 5.04, (GraphPad, San Diego, US). Integrated graphical views were generated using Circos plot [44]. The transcriptomics data were supplied as experiment GSE60815 at Gene Expression Omnibus according to MIAME regulations. All differentially expressed genes with a $\log_2$ fold change >1.5 and adjusted *p*-value <0.05 of each condition are supplied in Additional file 2.

qRT-PCR

For generating cDNA, a 1 μg RNA aliquot was used and reversely transcribed by the RevertAid Premium First Strand cDNA Synthesis Kit and random hexamer primers according to the manufacturer's instructions (Fermentas). Additionally, 1 μg of total RNA was used as RT- negative control following the same protocol with additional reaction buffer instead of the enzyme mix. Resulting cDNAs as well as RT-negative controls were diluted 1:50 in DNase- and RNase-free water. 1 μl of each sample was used for qRT-PCR.

Specific oligonucleotide primer pairs were used for PCR (Table 6). New primers or new primer pairs were designed with Primer3 software (http://frodo.wi.mit.edu/) and synthesized (Metabion, Martinsried, Germany). The amounts of cDNA of all genes were determined by qRT-PCR assays in 12.5 μl reaction volume. Conditions for the reactions were: 6.25 μl of 2× SsoFast Eva Green Supermix (BioRad, Hercules, US), 0.5 μM of each primer, 1 μl of cDNA; 1 × 95 °C for 3 min, 45 × 95 °C for 10 s and 57 °C for 15 s in a BioRad C1000 cycler with an CFX96 optical head. Validation of specific products was done via melting curve analysis, consisting of an initial heating at 95 °C for 10 s, followed by a stepwise temperature increase from 68 to 88 °C with an increment of 0.2 °C for 5 s. Threshold cycle values were calculated via regression analysis using CFX manager V 2.0 (BioRad). Differentially expressed genes were identified and analysed with the option 'gene study' of CFX manager software. The genes *pvu*A, *dna*E, *rec*A and a locus of the 16S-23S intergenic spacer region (1623S) were used for normalization via ΔΔC(q)-method.

Availability of supporting data

The transcriptomics data were supplied as experiment GSE60815 at Gene Expression Omnibus according to MIAME regulations. Available at: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE60815.

Additional files

Additional file 1: DAVID categories. Additional file includes DAVID data for all temperatures. (XLSX 25 kb)

Additional file 2: Differentially expressed genes with $\log_2$ fold change >1.5 and adjusted *p*-value <0.05 in each condition. Additional file includes the homolog and antagonistic reacting genes. (XLSX 192 kb)

Abbreviations

DAVID: Database for Annotation, Visualization and Integrated Discovery; T3SS-1: Type three secretion system one; T6SS: Type six secretion system; TDH: Thermostable direct hemolysin; TRH: Thermostable direct hemolysin related hemolysin.

Competing interests

The authors declare that they have no interests in competition.

Authors' contributions

SU conducted and performed the RT-PCR, qRT-PCR and microarray experiments. TA and STH participated in study design, data analysis, manuscript drafting and editing. AT, SAH and RA assisted with data analysis and manuscript revisions. All authors read and approved the manuscript.

Acknowledgements
We acknowledge Kathrin Oeleker for assistance in performing strain cultivations. The project was funded by the German Ministry of Education and Research (BMBF) within the VibrioNet project.

Author details
[1]Institute of Food Hygiene, Freie Universität Berlin, Berlin, Germany. [2]Department of Chemistry and Bioengineering, Tampere University of Technology, Tampere, Finland. [3]Department of Signal Processing, Tampere University of Technology, Tampere, Finland. [4]School of Health Sciences, University of Tampere, Tampere, Finland.

Received: 20 February 2015 Accepted: 12 October 2015
Published online: 23 October 2015

References

1. Potasman I, Paz A, Odeh M. Infectious outbreaks associated with bivalve shellfish consumption: a worldwide perspective. Clin Infect Dis. 2002;35:921–8.
2. Gopal S, Otta SK, Kumar S, Karunasagar I, Nishibuchi M. The occurrence of *Vibrio* species in tropical shrimp culture environments; implications for food safety. Int J Food Microbiol. 2005;102:151–9.
3. Slayton RB, Newton AE, Depaola A, Jones JL, Mahon BE. Clam-associated vibriosis, USA. Epidemiol Infect. 1988-2010;2013:1–6.
4. Vieira RH, de Sousa OV, Costa RA, Theophilo GN, Macrae A, et al. Raw oysters can be a risk for infections. Braz J Infect Dis. 2010;14:66–70.
5. Su YC, Liu C. *Vibrio parahaemolyticus*: a concern of seafood safety. Food Microbiol. 2007;24:549–58.
6. McLaughlin JB, DePaola A, Bopp CA, Martinek KA, Napolilli NP, et al. Outbreak of *Vibrio parahaemolyticus* gastroenteritis associated with Alaskan oysters. N Engl J Med. 2005;353:1463–70.
7. Chiang ML, Ho WL, Chou CC. Response of *Vibrio parahaemolyticus* to ethanol shock. Food Microbiol. 2006;23:461–7.
8. Chang CM, Chiang ML, Chou CC. Response of heat-shocked *Vibrio parahaemolyticus* to subsequent physical and chemical stresses. J Food Prot. 2004;67:2183–8.
9. Chiang ML, Chou CC. Survival of *Vibrio parahaemolyticus* under environmental stresses as influenced by growth phase and pre-adaptation treatment. Food Microbiol. 2009;26:391–5.
10. Phadtare S, Alsina J, Inouye M. Cold-shock response and cold-shock proteins. Curr Opin Microbiol. 1999;2:175–80.
11. Weber MH, Marahiel MA. Bacterial cold shock responses. Sci Prog. 2003;86:9–75.
12. Phadtare S. Recent developments in bacterial cold-shock response. Curr Issues Mol Biol. 2004;6:125–36.
13. Yang L, Zhou D, Liu X, Han H, Zhan L, et al. Cold-induced gene expression profiles of *Vibrio parahaemolyticus*: a time-course analysis. FEMS Microbiol Lett. 2009;291:50–8.
14. Gophna U, Ron EZ. Virulence and the heat shock response. Int J Med Microbiol. 2003;292:453–61.
15. Segal G, Ron EZ. Regulation of heat-shock response in bacteria. Ann N Y Acad Sci. 1998;851:147–51.
16. Yura T, Nagai H, Mori H. Regulation of the heat-shock response in bacteria. Annu Rev Microbiol. 1993;47:321–50.
17. Bukau B. Regulation of the *Escherichia coli* heat-shock response. Mol Microbiol. 1993;9:671–80.
18. Schroder H, Langer T, Hartl FU, Bukau B. DnaK, DnaJ and GrpE form a cellular chaperone machinery capable of repairing heat-induced protein damage. EMBO J. 1993;12:4137–44.
19. Wong HC, Peng PY, Lan SL, Chen YC, Lu KH, et al. Effects of heat shock on the thermotolerance, protein composition, and toxin production of *Vibrio parahaemolyticus*. J Food Prot. 2002;65:499–507.
20. Chiang ML, Chou CC. Expression of superoxide dismutase, catalase and thermostable direct hemolysin by, and growth in the presence of various nitrogen and carbon sources of heat-shocked and ethanol-shocked *Vibrio parahaemolyticus*. Int J Food Microbiol. 2008;121:268–74.
21. Mahoney JC, Gerding MJ, Jones SH, Whistler CA. Comparison of the pathogenic potentials of environmental and clinical *Vibrio parahaemolyticus* strains indicates a role for temperature regulation in virulence. Appl Environ Microbiol. 2010;76:7459–65.
22. Maurelli AT, Blackmon B, Curtiss R. 3rd Temperature-dependent expression of virulence genes in *Shigella* species. Infect Immun. 1984;43:195–201.
23. Straley SC, Perry RD. Environmental modulation of gene expression and pathogenesis in *Yersinia*. Trends Microbiol. 1995;3:310–7.
24. Rappuoli R, Arico B, Scarlato V. Thermoregulation and reversible differentiation in *Bordetella*: a model for pathogenic bacteria. Mol Microbiol. 1992;6:2209–2211.35.
25. Gonzalez-Escalona N, Martinez-Urtaza J, Romero J, Espejo RT, Jaykus LA, et al. Determination of molecular phylogenetics of *Vibrio parahaemolyticus* strains by multilocus sequence typing. J Bacteriol. 2008;190:2831–40.
26. Zhu J, Shimizu K. The effect of pfl gene knockout on the metabolism for optically pure D-lactate production by *Escherichia coli*. Appl Microbiol Biotechnol. 2004;64(3):367–75.
27. Fields PA. Review: Protein function at thermal extremes: balancing stability and flexibility. Comp Biochem Physiol A Mol Integr Physiol. 2001;129(2–3):417–31.
28. Badger JL, Miller VL. Role of RpoS in survival of Yersinia enterocolitica to a variety of environmental stresses. J Bacteriol. 1995;177:5370–3.
29. Persson BC, Jäger G, Gustafsson C. The spoU gene of Escherichia coli, the fourth gene of the spoT operon, is essential for tRNA (Gm18) 2′-O-methyltransferase activity. Nucleic Acids Res. 1997;25(20):4093–7.
30. Wong SM, Akerley BJ. Environmental and genetic regulation of the phosphorylcholine epitope of *Haemophilus influenzae* lipooligosaccharide. Mol Microbiol. 2005;55(3):724–38.
31. Vorachek-Warren MK, Carty SM, Lin S, Cotter RJ, et al. An *Escherichia coli* mutant lacking the cold shock-induced palmitoleoyltransferase of lipid A biosynthesis: absence of unsaturated acyl chains and antibiotic hypersensitivity at 12 degrees C. J Biol Chem. 2002;277(16):14186–93.
32. Makino K, Oshima K, Kurokawa K, Yokoyama K, Uda T, et al. Genome sequence of *Vibrio parahaemolyticus*: a pathogenic mechanism distinct from that of *V. cholerae*. Lancet. 2003;361:743–9.
33. Ma L, Zhang Y, Yan X, Guo L, Wang L, et al. Expression of the type VI secretion system 1 component Hcp1 is indirectly repressed by OpaR in *Vibrio parahaemolyticus*. ScientificWorldJournal. 2012;982140.
34. Somerville GA, Proctor RA. At the crossroads of bacterial metabolism and virulence factor synthesis in Staphylococci. Microbiol Mol Biol Rev. 2009;73:233–48.
35. Salomon D, Gonzalez H, Updegraff BL, Orth K. *Vibrio parahaemolyticus* Type VI Secretion System 1 is activated in marine conditions to target bacteria, and is differentially regulated from System 2. PLoS One. 2013;8, e61086.
36. Nasu H, Iida T, Sugahara T, Yamaichi Y, Park KS, et al. A filamentous phage associated with recent pandemic *Vibrio parahaemolyticus* O3:K6 strains. J Clin Microbiol. 2000;38:2156–61.
37. Thompson FL, Cleenwerck I, Swings J, Matsuyama J, Iida T. Genomic diversity and homologous recombination in *Vibrio parahaemolyticus* as revealed by amplified fragment length polymorphism (AFLP) and multilocus sequence analysis (MLSA). Microbes Environ. 2007;22:373–9.
38. Bolstad BM, Irizarry RA, Astrand M, Speed TP. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics. 2003;19:185–93.
39. Smyth GK. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol. 2004;3(1):Art 3.
40. Benjamini Y, Yekutieli D. The control of the false discovery rate in multiple testing under dependency. Ann Stat. 2001;4:1165–88.
41. da Huang W, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat Protoc. 2009;4:44–57.
42. da Huang W, Sherman BT, Lempicki RA. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res. 2009;37:1–13.
43. Sturn A, Quackenbush J, Trajanoski Z. Genesis: cluster analysis of microarray data. Bioinformatics. 2002;18:207–8.
44. Krzywinski MI, Schein JE, Birol I, Connors J, Gascoyne R, et al. Circos: An information aesthetic for comparative genomics. Genome Res. 2009;19(9):1639–45.
45. Kong RY, Lee SK, Law TW, Law SH, Wu RS. Rapid detection of six types of bacterial pathogens in marine waters by multiplex PCR. Water Res. 2002;36:2802–12.

Publication III

Syeda Sakira Hassan, Pekka Ruusuvuori, Leena Latonen, and Heikki Huttunen. "Flow Cytometry-Based Classification in Cancer Research: A View on Feature Selection", *Cancer informatics*, doi: 10.4137/CIN.S30795, vol 14, no. S(5), pp. 75-85, Feb. 2016.

Flow Cytometry-Based Classification in Cancer Research: A View on Feature Selection

S. Sakira Hassan[1], Pekka Ruusuvuori[2,3], Leena Latonen[3] and Heikki Huttunen[1]

[1]Department of Signal Processing, Tampere University of Technology, Tampere, Finland. [2]Pori Department, Tampere University of Technology, Pori, Finland. [3]BioMediTech, University of Tampere, Tampere, Finland.

Supplementary Issue: Statistical Systems Theory in Cancer Modeling, Diagnosis, and Therapy

ABSTRACT: In this paper, we study the problem of feature selection in cancer-related machine learning tasks. In particular, we study the accuracy and stability of different feature selection approaches within simplistic machine learning pipelines. Earlier studies have shown that for certain cases, the accuracy of detection can easily reach 100% given enough training data. Here, however, we concentrate on simplifying the classification models with and seek for feature selection approaches that are reliable even with extremely small sample sizes. We show that as much as 50% of features can be discarded without compromising the prediction accuracy. Moreover, we study the model selection problem among the ℓ_1 regularization path of logistic regression classifiers. To this aim, we compare a more traditional cross-validation approach with a recently proposed Bayesian error estimator.

KEYWORDS: AML, leukemia, flow cytometry, logistic regression, error estimation, model selection

SUPPLEMENT: Statistical Systems Theory in Cancer Modeling, Diagnosis, and Therapy

CITATION: Hassan et al. Flow Cytometry-Based Classification in Cancer Research: A View on Feature Selection. *Cancer Informatics* 2015:14(S5) 75–85 doi: 10.4137/CIN.S30795.

TYPE: Original Research

RECEIVED: November 18, 2015. **RESUBMITTED:** February 01, 2016. **ACCEPTED FOR PUBLICATION:** February 07, 2016.

ACADEMIC EDITOR: J. T. Efird, Editor in Chief

PEER REVIEW: Six peer reviewers contributed to the peer review report. Reviewers' reports totaled 1860 words, excluding any confidential comments to the academic editor.

FUNDING: Authors disclose no external funding sources.

COMPETING INTERESTS: Authors disclose no potential conflicts of interest.

CORRESPONDENCE: sakira.hassan@tut.fi

Paper subject to independent expert blind peer review. All editorial decisions made by independent academic editor. Upon submission manuscript was subject to anti-plagiarism scanning. Prior to publication all authors have given signed confirmation of agreement to article publication and compliance with all applicable ethical and legal requirements, including the accuracy of author and contributor information, disclosure of competing interests and funding sources, compliance with ethical requirements relating to human and animal study participants, and compliance with any copyright requirements of third parties. This journal is a member of the Committee on Publication Ethics (COPE).

Published by Libertas Academica. Learn more about this journal.

Introduction

Flow cytometry enables quantitative measurement of single-cell properties through visible and fluorescent light in a high-throughput manner. The measured signals include fluorescence emission and light scatter. Flow cytometry has been routinely used for detecting malignancies from blood samples.[1] Through technological advances, measuring the combination of fluorescent signals from several different channels has enabled the use of high-dimensional data for studies, such as cytometric fingerprinting[2] and large-scale analysis of cell types.[3]

In this paper, we study the analysis of flow cytometry data from the feature selection point of view. More specifically, flow cytometry is able to produce large quantities of partially redundant measurement data, and the selection of important quantities within the large body of measurements is of interest. Moreover, a typical scenario contains large quantities of measurement data but may be limited to only a few patients. Thus, an ideal method would distil only the essential parts of the measurements from each patient, while producing reliable and well generalizing results when only a small amount of individuals is available in the training data.

We will concentrate our attention on two particular sets of flow cytometry data. The first set originates from the acute myeloid leukemia (AML) prediction challenge of the DREAM initiative in 2013.[4] The competition attracted a number of teams, and as several researchers use the data as part of their work, the challenge data have become a standard benchmark within the field. For example, Aghaeepour et al.[4] presents a large pool of analysis approaches from the DREAM challenge. Among classification methods presented in the literature, there are several sophisticated machine learning approaches, such as learning vector quantization,[5] correlative matrix mapping, and relative entropy differences.[6] The strength of data-driven approaches relying on supervised classification is their ability to handle high-dimensional data without requiring prior knowledge of the biological application.

The DREAM AML data represent a relatively large-scale experiment consisting of altogether 179 patients. Although it is a small number for traditional machine learning problems, the number of patients is unusually large for a biological study. To this aim, we use another dataset that represents a more commonly encountered sample size of 16 samples extracted from a prostate cancer cell line. This dataset, with two different treatments and a low number of samples, presents a nontrivial but common challenge for prediction and related feature selection. More information on the two datasets is provided in Data section.

In our earlier work,[7] we presented a supervised classification pipeline based on linear discriminant analysis (LDA) and logistic regression (LR) classifiers. Briefly, the method first

transforms the measurement data into higher dimensional space by generating combined features with multiplications and divisions between measurements. Following this mapping into higher dimensional space, LDA is used for lowering the dimensionality into a single value per measurement. Then, empirical distribution functions (EDFs) are constructed from LDA results for AML-positive and AML-negative sample classes and compared to training EDFs of both classes. The comparison results in two similarity values per group of measurements, and these results are fed to the LR classifier for a final AML prediction result. Our approach, together with alternative well-performing approaches,[4,5] represents a relatively complicated pipeline of somewhat arbitrary computation steps. Thus, our interest is to simplify these pipelines into a simple collection of obvious features, while still retaining a good accuracy.

Our approach here is to use LR classifier applied to summarize features, which are the mean and standard deviation of the measurements instead of the complete data. This reduces the number of features used in classification and, subsequently, also the model complexity. An essential part of classifier design is error estimation, which guides model selection.[8] Our strategy for model selection is to apply the recently introduced Bayesian error estimator (BEE).[9] We compare BEE model selection with a traditional 10-fold cross-validation (CV-10) error estimation, as well as with Bayesian information criterion (BIC)-based model selection, and conclude that the proposed approach enables accurate prediction for flow cytometry data with fewer measurements and a less complex classifier model than those previously presented in the literature.

The rest of this paper is organized as follows. In Materials and Methods section, we describe the data and methods used in this study and briefly discuss how feature selection is commonly done in machine learning. Experimental Results section presents the results of our experiments with different model and feature selection criteria for the materials introduced in Materials and Methods section. Finally, in Conclusions section, we summarize the work and discuss the conclusions of the results.

Materials and Methods

In this section, we describe the datasets used in this paper. We also give a brief overview of modeling method for feature selection. In addition to this, we introduce the state-of-the-art Bayesian error estimator (BEE) for model parameter selection. Finally, we present an example where the performance of BEE is benchmarked against other model selection criteria.

Data. In this work, we study two datasets: A larger set with 179 samples and a smaller set with 16 samples. These two case studies represent different classification challenges in terms of both application and sample size.

AML dataset. The flow cytometry dataset for the AML experiment has been collected from the DREAM6-FlowCAP2 challenge, which was organized by the DREAM project and the FlowCAP initiative (DREAM challenge AML dataset can be accessed from Aghaeepour et al).[4] We use the training dataset that consists of flow cytometry measurements of 179 patients. Among them, 23 patients are AML positive and the remaining 156 patients are AML negative. The flow cytometry measurement for each patient corresponds to seven jointly measured groups (hereafter called tubes) of seven quantities with a total of 49 biomarker measurements per cell. The biomarkers are summarized in Table 1 and include *Forward Scatter* in linear scale (FS Lin), *Sideward Scatter* in logarithmic scale (SS Log), and five fluorescence intensities (FL1–FL5) in logarithmic scales. For calibration purposes, FS Lin, SS Log, and CD45-ECD were measured for all tubes and the other 28 biomarkers were measured only in one tube.

Cancer cell line dataset. As another case study, we use flow cytometry data from a small sample setting. The data come from a prostate cancer cell line 22Rv1 stained with propidium iodide for cell cycle analysis.[10] The cells are transfected with miRNAs (either control or miR-193b) and induced to proliferate by overexpression of cyclin D. The data consist of 16 samples, with 8 samples (without cyclin D overexpression) with relatively consistent cell cycle profile and 8 samples (overexpressing cyclin D) with an altered cell cycle profile, ie, induced cell cycle activity with an increase in cells in DNA synthesis phase. The samples for both classes include four repetitions of two treatments, which are considered here to represent the same class. Each sample contains 12 measured channels, consisting of two scatter measurements and four fluorescence channels, both as area and height measurements.

Table 1. List of seven tubes with biomarkers provided in DREAM6 AML prediction data.

			FL1 Log	FL2 Log	FL3 Log	FL4 Log	FL5 Log
tube 1	fs Lin	ss Log	IgG1-fItC	IgG1-Pe	CD45-eCD	IgG1-PC5	IgG1-PC7
tube 2	fs Lin	ss Log	Kappa-fIt	Lambda-Pe	CD45-eCD	CD19-PC5	CD20-PC7
tube 3	fs Lin	ss Log	CD7-fItC	CD4-Pe	CD45-eCD	CD8-PC5	CD2-PC7
tube 4	fs Lin	ss Log	CD15-fItC	CD13-Pe	CD45-eCD	CD16-PC5	CD56-PC7
tube 5	fs Lin	ss Log	CD14-fItC	CD11c-Pe	CD45-eCD	CD64-PC5	CD33-PC7
tube 6	fs Lin	ss Log	HLa-Dr -fItC	CD117-Pe	CD45-eCD	CD34-PC5	CD38-PC7
tube 7	fs Lin	ss Log	CD5-fItC	CD19-Pe	CD45-eCD	CD3-PC5	CD10-PC7

Feature extraction. Several feature extraction methods can be used to obtain meaningful features from raw flow cytometry measurements. For instance, among widely used feature extraction techniques are methods based on principal component analysis and histogram computation. Biehl et al proposed statistical divergences to extract features that include moments, median, and interquartile range.[5] The length of the feature vector was 186 in this case. Another well-performed model was based on multidimensional entropic distance-based features.[4,5] Manninen et al.[7] expanded the cell measurements of each tube to a higher dimensional space. Following this transformation, LDA is used to lower the dimensionality into a single value for each measurement. These previous studies are summarized in Table 2. Table 2 also tabulates the test accuracy in terms of the area under the receiver operating characteristics (ROC) curve (AUC) measure over a single train/test split, which should not be interpreted as a definitive measure of accuracy, as the split of the samples is just one instance of all possible splits.

Table 2. Studies based on feature extraction strategies for DREAM amL challenge dataset.

	ACCURACY	SIzE o F FEATURE VECTo R	BRIEF DESCRIPTIo N
Biehl et al.[5]	1.00	186	Extraction of features with moments, median and interquartile and learning vector quantization is used for prediction
Vilar et al.[4]	1.00	31	Extraction of features with entropies and histogram based classi□er is used for prediction
manninen et al.[7]	1.00	(# of events) x 84	Expand features to higher dimension and then mapping to 1-D using linear discriminant analysis; logistic regression is used for prediction
o ur solution (this study)	0.9989	49	Extraction of feature vector from means of measurements and applying regularized logistic regression for prediction
o ur solution (this study)	0.9992	98	Extraction of feature vector from means and standard deviation of measurements and applying regularized logistic regression for prediction

Note: The accuracy is measured in terms of the AUC of a single train/test split.

In this paper, we use one of the simplest feature extraction techniques that include only the mean and the standard deviation of the each measurement. For the first dataset, the length of this extracted feature vector is 98, comprising 49 mean values and 49 standard deviations. As seen in the experiments of Experimental Results section, these features are sufficient to separate the classes without compromising the prediction accuracy. We will consider two versions of these basic features: the first feature set contains only the 49 mean values of the measurements, while the second feature vector considers both mean values and standard deviations, with altogether 98 features. The same approach is used with the smaller dataset, thus producing two different experimental cases. Before training the classifiers, we normalized all features to the interval (0, 1).

LR and regularization. LR is a discriminative method for modeling the class conditional probability densities by the logistic function. Given an observation matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$ with N observations, P features, and corresponding class labels $\mathbf{Y} \in 1,\ldots,C$, we define LR model for the binary classification as,

$$p(y = 1 \mid x) = \frac{\exp(\beta_0 + \boldsymbol{\beta}^T x)}{1 + \exp(\beta_0 + \boldsymbol{\beta}^T x)} \quad (1)$$

and

$$p(y = 0 \mid x) = 1 - p(y = 1 \mid x) = \frac{1}{1 + \exp(\beta_0 + \boldsymbol{\beta}^T x)}. \quad (2)$$

Here, $\boldsymbol{x}$ represents a feature vector in the feature space corresponding to class label $y \in \{0,1\}$, β_0 is the intercept, and $\boldsymbol{\beta}$ represents coefficients of the logistic model. We can determine the model parameters β_0 and $\boldsymbol{\beta}$ from the training dataset by solving the ℓ_1 penalized LR problem,

$$\arg\min_{\beta_0, \boldsymbol{\beta}} \sum_{i=1}^{N} -\log p(y_i \mid \mathrm{x}_i) + \lambda \| \boldsymbol{\beta} \|_1, \quad (3)$$

where $\lambda > 0$ is the regularization parameter. When the number of training data is not larger compared to the number of features, ie, $P \gg N$, regularization is used to solve the overfitting problem.[11] In regularization, an extra term, λ, is added, which controls the trade-off between the loss function and the size of the coefficients. More recently, in feature selection, ℓ_1-regularized LR has received much attention, as it yields a sparse solution that has relatively few nonzero coefficients.[12] This minimization task is analogous to *least absolute shrinkage and selection operator* (Lasso) algorithm proposed by Tibshirani.[13] In addition to this, several extensions of Lasso have also been developed, such as grouped Lasso,[14,15] Dantzig selector,[16] elastic net,[17] and graphical Lasso.[18] In this paper,

we use the GLMNET algorithm by Friedman et al.[19] that combines the ℓ_2 and ℓ_1 penalties:

$$\arg\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{N} -\log p\left(y_i \mid \mathbf{x}_i\right) + \lambda\left(\alpha \parallel \boldsymbol{\beta} \parallel_1 + (1-\alpha) \parallel \boldsymbol{\beta} \parallel_2\right), \quad (4)$$

where $\lambda > 0$ and $\alpha \in [0,1]$. The parameter α is a compromise between the ℓ_1 and ℓ_2 penalties, thereby determining the type of regularization. On the other hand, the regularization parameter λ controls the amount of regularization. A very large λ will completely shrinks the coefficients to zero and may yield a null or empty model.

In general, the model parameters λ and α are selected using the CV approach.[20] The dataset is randomly split into K mutually exclusive subsets of approximately equal sized. In K-fold CV, the process is iterated k times. At the kth iteration, the Kth fold is retained as test set and the remaining K − 1 folds are used as training set to train the model. Each of the K-folds is tested exactly once. The test set assesses the quality of the trained model. Then, the K results are combined or averaged to produce a single estimation of the model. The most commonly used values for K are 5 and 10. In this experiment, we set the value of $\alpha = 1$ and CV-10 is used for the selection of the model parameter λ and assessment of the model. As the type of regularization is determined by α, setting $\alpha = 0$ provides ℓ_2 penalty that is useful in cases, where the features are mutually correlated. On the other hand, $\alpha = 1$ provides sparse solution with fewer coefficients and, in turn, this is suitable for implicit feature selection. We have also experimented with 5-fold CV, but the results do not improve significantly.

Bayesian error estimator. A Bayesian approach to error estimation was recently introduced in the context of discrete classifiers[21] and linear classifiers.[22] The Bayesian error estimator (BEE) estimates the classification error directly from the training set and has shown to improve both the accuracy and speed of the actual error estimate[21,22] compared to traditional counting-based approaches, such as CV. In our earlier papers, we have shown that BEE has improved the stability and speed of computation in the model selection context as well.[9,23] We will next briefly review the definition of BEE for a fixed linear two-class classifier specified by the parameters $\boldsymbol{\beta}$ and β_0.

The Bayesian error estimator for linear classification assumes that the samples from each class are independent and identically distributed Gaussian random variables. For the two classes, the parameters (mean and covariance) of the Gaussian model are denoted as $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ and the corresponding priors for the parameters are denoted as $p_0(\boldsymbol{\theta})$ and $p_1(\boldsymbol{\theta})$. Then, the posterior probability density functions (PDFs) of parameters for class $c \in \{0,1\}$ are given by the Bayes' rule:

$$p_c^*\left(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}\right) \propto p_c(\boldsymbol{\theta}) \prod_{i:\, y_i = c} p_c\left(\mathbf{x}_i \mid \boldsymbol{\theta}\right), \quad (5)$$

where $p_c\left(\mathbf{x}_i \mid \theta\right)$ is the Gaussian class conditional density of $c \in \{0,1\}$.

The Bayesian error estimator (BEE) is defined as the minimum mean squared estimator by minimizing the expectation between the error estimate and the true error. This quantity is composed of class-specific conditional expected errors balanced by the priors $p(c)$ for the two classes $c \in \{0,1\}$[22]:

$$\mathrm{BEE} \triangleq E\left[\varepsilon \mid \mathbf{X}, \mathbf{y}\right] = p(0)E\left[\varepsilon_0 \mid \mathbf{X}, \mathbf{y}\right] + p(1)E\left[\varepsilon_1 \mid \mathbf{X}, \mathbf{y}\right], \quad (6)$$

with the expected classification error of samples from class c given by

$$E\left[\varepsilon_c \mid \mathbf{X}, \mathbf{y}\right] = \int \varepsilon_c(\boldsymbol{\theta}) p_c\left(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}\right) d\boldsymbol{\theta}, \quad (7)$$

where $\varepsilon_c(\theta)$ denotes the true classification error.

The integral of Equation (7) can be evaluated by assuming an inverse Wishart prior for the class conditional density:

$$p_c(\boldsymbol{\theta}) \propto \det\left(\boldsymbol{\Sigma}_c\right)^{-(\kappa+P+1)/2} \exp\left(\frac{1}{2}\mathrm{trace}\left(\mathbf{S}\boldsymbol{\Sigma}_c^{-1}\right)\right) \times \det\left(\boldsymbol{\Sigma}_c\right)^{-(1/2)} \exp\left(-(\upsilon/2)\left(\boldsymbol{\mu}_c - \mathbf{m}\right)^T \boldsymbol{\Sigma}_c^{-1}\left(\boldsymbol{\mu}_c - \mathbf{m}\right)\right), \quad (8)$$

where $\upsilon \in \mathbb{R}, \kappa \in \mathbb{R}, \mathbf{S} \in \mathbb{R}^{P \times P}$, and $\mathbf{m} \in \mathbb{R}$ are the hyperparameters of the Bayesian model and $\boldsymbol{\theta}_c = \left(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right)$ is the parameter of the Gaussian distribution. Different choices of the values of hyperparameters lead to different error estimators, but we will concentrate on a specific choice shown to be successful in earlier works[9,22]: $\kappa = P + 2$, $\upsilon = 0.5$, $\mathbf{S} = \mathbf{I}$, and $\mathbf{m} = \mathbf{0}$. For the resulting simplified closed-form solution, refer to Ref.[9] Matlab and Python implementations of BEE are available for download (https://sites.google.com/site/bayesianerrorestimate/).

Model selection. Model selection is a critical aspect in classifier design. Moreover, most modern classifiers are tuned by a set of hyperparameters, whose selection has a substantial effect on the resulting accuracy as well. Thus, the selection of an appropriate model family and the associated hyperparameters requires an accurate measure for comparing the accuracies of the model candidates. In our work, we are primarily interested in the selection of the regularization parameter λ of an LR classifier. However, it is to be noted that the methodology applies to any linear classifier.

The prediction accuracy and selection of the best model can be quantified by error estimators. CV estimator is often used to select the best value of the model selection parameter λ along a regularization path. As an example, error curves for different values of λ are illustrated in Figure 1. For this purpose, we used the flow cytometry training data of 49 features and 179 observations. For an individual tube, each feature represents the average of the biomarker intensities. The error curves are estimated for different values of λ ranging from 10^{-9} to 10^{0}.

In the example of Figure 1 (left panel), a 5-fold CV procedure is repeated 100 times and each step includes five training iterations on partial data. The error curves obtained for 100 iterations of 5-fold CV illustrate the significant deviation

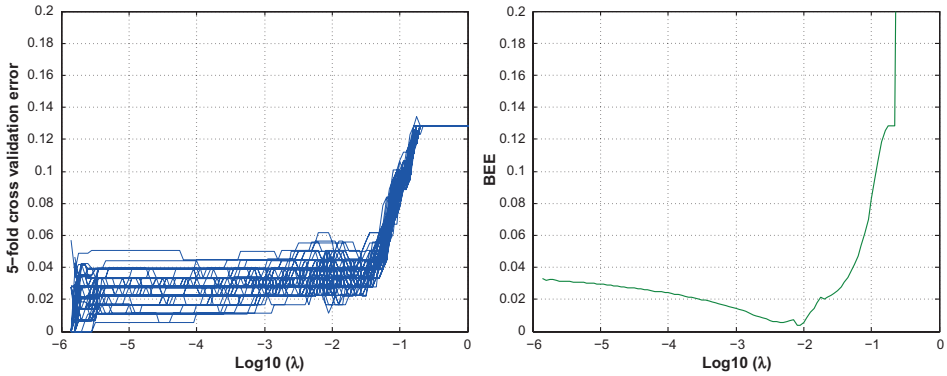


Figure 1. Left: Examples of regularization path error curves of 5-fold CV for ou□ ow cytometry data with healthy and AML positive classes. Right: The corresponding Bee curve.

of the regularization paths from one iteration to another. The deviation is due to the randomness in splitting the training data into folds, which results in an individual error estimate for each split. Moreover, for a very small number of samples, such as 5 or 10, the split to validation and training sets for the K-fold CV estimator may not be appropriate. In fact, in this experiment, the K-fold CV approach fails to estimate the errors for smaller λ, as the number of samples split by CV is insufficient. On the other hand, Figure 1 (right panel) illustrates the error estimate of BEE, which is a single deterministic error curve. It is to be noted that the curve recognizes model overfitting (error estimate starts to increase for small regularization terms λ), although the error is estimated directly from the training set. No splitting or iterative resampling is required, which in turn accelerates the computation.

Experimental Results

In the following section, we present the experimental results. First, we demonstrate different model selection criteria to estimate the significant features. Then, we assess the performances of those methods in the AML classification case. Finally, we present the results for the second, small sample case.

Comparison of model selection criteria. Typical approach for the selection of model parameter is CV.[13] In this paper, we also consider Bayesian error estimator (see Bayesian Error Estimator section) and BIC[24] as alternative approaches to estimate the regularized parameter. In order to study the behavior of different parameter selection criteria, we first train the LR classifier with the training data along the decreasing sequence of regularization path with log10 (λ)∈{0,–0.05,–0.1, –0.15,…, –8.90, –8.95, –9.00}. Then, again the whole training data are used to estimate the error rate for each λ. Finally, for each estimator, we select the model with λ value that achieves the minimum error rate. As resampling in CV-10 introduces randomness, in this case, we iterate 200 times and the result is averaged. The deterministic nature in BEE and BIC will produce the same result on the training data at each iteration.

The results are summarized in Table 3. For all methods, minimum error rates, AUC, and the number of selected features are estimated from the whole training data. It is to be noted that the reported AUC is computed from the training to emphasize that all feature sets are enough to partition the feature space into two categories perfectly. The test error is reported later.

The results indicate that the number of features selected by BEE method is lower compared to those of CV and BIC. For the first feature vector with size 49, BEE selects only 14 features as significant, while for the second feature vector with

Table 3. Parameter selection by different estimators: average number of selected features, λ, aUC, and their standard deviations with training data.

METho D	FEATURE TYPE	NUMBER oF SELECTED FEATURES	SELECTED Log10 (λ)	AUC (TRAINING)
CV-10	mean	19.72 ± 2.41	–2.95 ± 2.30	0.9997 ± 0.0017
CV-10	mean and std	23.91 ± 0.80	–4.14 ± 3.00	1 ± 0.0000
Bee	mean	15 ± 0.00	–2.05 ± 0.00	0.9989 ± 0.0000
Bee	mean and std	13 ± 0.00	–1.80 ± 0.00	0.9992 ± 0.0000
BIC	mean	20 ± 0.00	–5.85 ± 0.00	1 ± 0.0000
BIC	mean and std	24 ± 0.00	–5.70 ± 0.00	1 ± 0.0000

length 98, BEE selects only 12 features. Tables 4 and 5 list the selected features, ie, significant biomarkers along with the corresponding coefficient values. Due to the randomness in CV-10, we only present the results of one iteration as an illustration: there is a significant variation of the selected features depending on the chosen CV split. However, it is to be noted that the coefficients of BIC and BEE are not specific to this particular iteration, as they do not include the random split.

Performance assessment of the model selection criteria. The performances of the model selection methods are studied in the following section. The classification error is considered as the performance criterion, and both false positives (healthy control classified as AML) and false negatives (AML classified as healthy control) are counted with equal weight. The performance of the Bayesian error estimator is benchmarked against those of CV-10 and BIC for a different number of sample sizes. For this purpose, a randomly selected proportion of 10%, 15%, 20%–90%, and 95% is selected for training the classifier, while the remaining data are used for performance assessment. For each training sample, the experiment is executed 200 times by generating a new training set each time.

Table 4. The nonzero coefficients of features with mean.

TUBE	FEATURE	10-Fold CV	BEE	BIC
	Constant	−13.38	−3.50	−13.54
tube 1	fs Lin	0.75	**0**	0.78
tube 1	ss Log	−5.92	−0.73	−6.01
tube 1	f L1:IgG1-f ltC	−0.46	−0.19	−0.46
tube 1	f L4:IgG1-PC5	−2.08	**0**	−2.13
tube 1	f L5:IgG1-PC7	−3.07	−0.19	−3.14
tube 2	fs Lin	0.001	**0**	**0**
tube 2	f L5:CD20-PC7	2.76	**0**	2.78
tube 3	ss Log	**0**	−0.77	**0**
tube 3	f L4:CD8-PC5	−1.94	−0.10	−1.97
tube 4	fs Lin	0.97	**0**	0.97
tube 4	f L1:CD15-f ltC	−4.77	**0**	−4.82
tube 4	f L2:CD13-Pe	3.44	0.21	3.48
tube 4	f L4:CD16-PC5	**0**	−0.09	**0**
tube 4	f L5:CD56-PC7	3.29	0.75	3.35
tube 5	fs Lin	2.35	**0**	2.40
tube 5	f L2:CD11c-Pe	−0.15	**0**	−0.16
tube 5	f L3:CD45-eCD	−1.84	−0.02	−1.84
tube 5	f L4:CD64-PC5	1.66	**0**	1.69
tube 5	f L5:CD33-PC7	0.75	0.60	0.76
tube 6	f L2:CD117-Pe	4.82	0.89	4.88
tube 6	f L4:CD34-PC5	6.88	0.72	6.99
tube 6	f L5:CD38-PC7	**0**	0.41	**0**
tube 7	f L1:CD5-f ltC	−4.68	−0.19	−4.74

Note: The size of the feature sets is 49, of which the CV, BEE, and BIC select 20, 14, and 19 features, respectively.

Classification errors. The error curves for different sample sizes are shown in Figure 2. The procedure is repeated 200 times for each training sample size, and the average of classification error is computed for each model selection criterion. With a very small number of training samples, such as 10% or 15% of the dataset, BEE provides improved accuracy over CV-10 and BIC (Fig. 2 left and right panels). For instance, with 10% training samples, the classification errors of the model selected by CV-10 are 7.56% (Fig. 2 left panel) and 7.41% (Fig. 2 right panel) higher than those of BEE. In case of BIC, the classification error is 7.81% higher than that of BEE (Fig. 2 left panel). As the number of training samples increases, for example, above 60%, the performance of BIC exceeds than that of BEE (Fig. 2 left panel). However, the performance of BIC is similar to that of BEE when more features are involved in the experiment (Fig. 2 right panel).

Table 5. The nonzero coefficients of features with mean and standard deviation.

TUBE	FEATURE		10-Fold CV	BEE	BIC
	Constant		−13.47	−3.27	−13.47
tube 1	fs Lin	mean	0.027	**0**	0.027
tube 1	ss Log	mean	−4.49	−0.47	−4.49
tube 1	f L1:IgG1-f ltC	std	−3.80	−0.22	−3.80
tube 1	f L5:IgG1-PC7	std	−1.60	−0.16	−1.60
tube 2	f L5:CD20-PC7	mean	0.50	**0**	0.50
tube 3	ss Log	mean	**0**	−0.29	**0**
tube 3	f L5:CD2-PC7	mean	−0.48	**0**	−0.48
tube 3	f L5:CD2-PC7	std	−1.21	**0**	−1.21
tube 4	fs Lin	mean	0.05	**0**	0.05
tube 4	f L1:CD15-f ltC	mean	−0.14	**0**	−0.14
tube 4	f L2:CD13-Pe	mean	2.50	**0**	2.50
tube 4	f L4:CD16-PC5	mean	−1.32	**0**	−1.32
tube 4	f L4:CD16-PC5	std	**0**	−0.39	**0**
tube 4	f L5:CD56-PC7	std	6.07	0.45	6.07
tube 5	f L1:CD14-f ltC	std	−0.004	**0**	−0.004
tube 5	f L3:CD45-eCD	mean	−2.51	**0**	−2.51
tube 5	f L5:CD33-PC7	mean	1.47	0.32	1.47
tube 5	f L5:CD33-PC7	std	−0.70	**0**	−0.70
tube 6	ss Log	std	**0**	−0.23	**0**
tube 6	f L2:CD117-Pe	mean	3.19	0.46	3.19
tube 6	f L2:CD117-Pe	std	2.19	**0**	2.19
tube 6	f L4:CD34-PC5	std	1.88	0.75	1.88
tube 6	f L5:CD38-PC7	mean	1.31	0.44	1.31
tube 7	fs Lin	mean	1.39	**0**	1.39
tube 7	fs Lin	std	**0**	−0.16	**0**
tube 7	f L1:CD5-f ltC	std	−1.16	**0**	−1.16
tube 7	f L5:CD10-PC7	std	0.04	**0**	0.04

Note: The size of the feature sets is 98, of which the CV, BEE, and BIC select 23, 12, and 23 features, respectively.

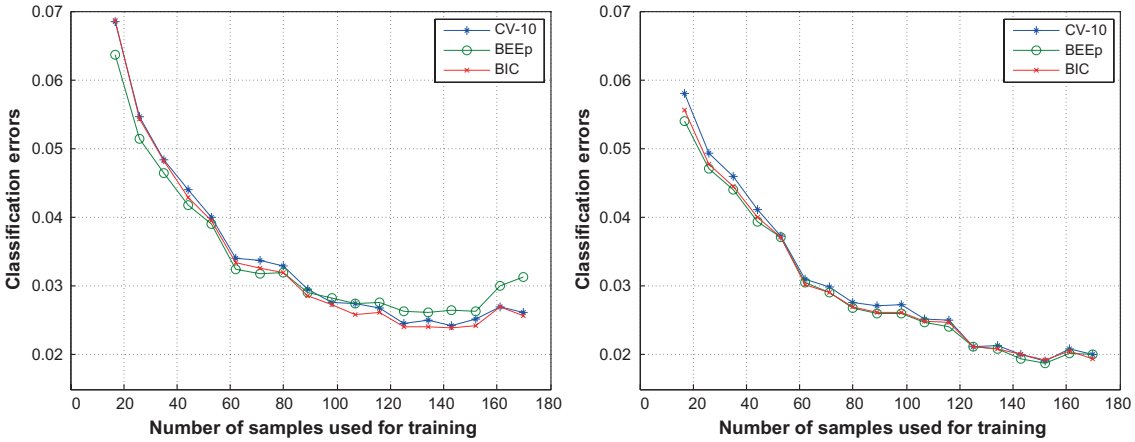


Figure 2. The average classification error curves for CV-10, BEE with proper prior (BEEp), and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements.

Area under the ROC curve. In this section, we evaluate the performance in terms of AUC. Figure 3 illustrates the average of AUC for different training sample sizes. Here, the BEE method achieves improvement over the other methods. With small training samples, for instance, 10%, the average AUC of BEE is 1.11% (Fig. 3 left panel) and 1.30% (Fig. 3 right panel) higher than that of CV-10. As the number of training samples increases, CV-10 and BIC also converge toward the results of BEE; however, the BEE selected model consistently results in the highest AUC score. With the larger feature vector that includes the measurements of mean and standard deviation, the average AUC curves of BEE and BIC follow the similar pattern (Fig. 3 right panel).

Number of selected features. We further assess the performances of the estimators using feature selection criteria. At each iteration, we determine the total number of selected features that have nonzero values for a different number of training samples. Then, we compute the average and the variability (ie, standard deviation) of the selected features for different training samples. The results are illustrated in Figure 4. For BEE, the average number of selected features is lower in amount compared to those of CV and BIC (Fig. 4 top panel). For instance, with 95% training samples, BEE requires 36.49% and 33.89% less features than CV and BIC, respectively, for model prediction (Fig. 4 top-right panel). Moreover, the variability in selected features using BEE is also comparable (Fig. 4 bottom panel). The CV-10 has the worst performance. Although BIC shows that the deviation in feature selection at different iterations is smaller, the

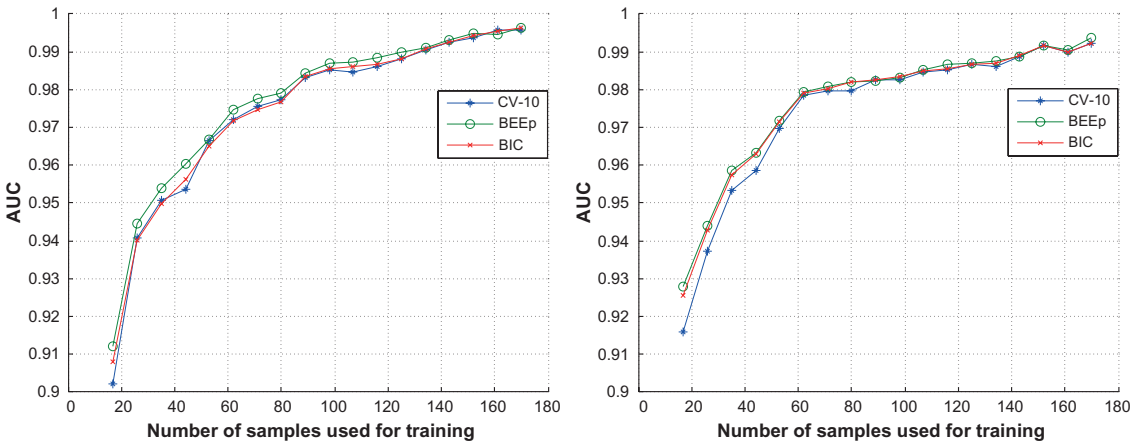


Figure 3. The average AUCs for CV-10, BEE with proper prior (BEEp), and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements.

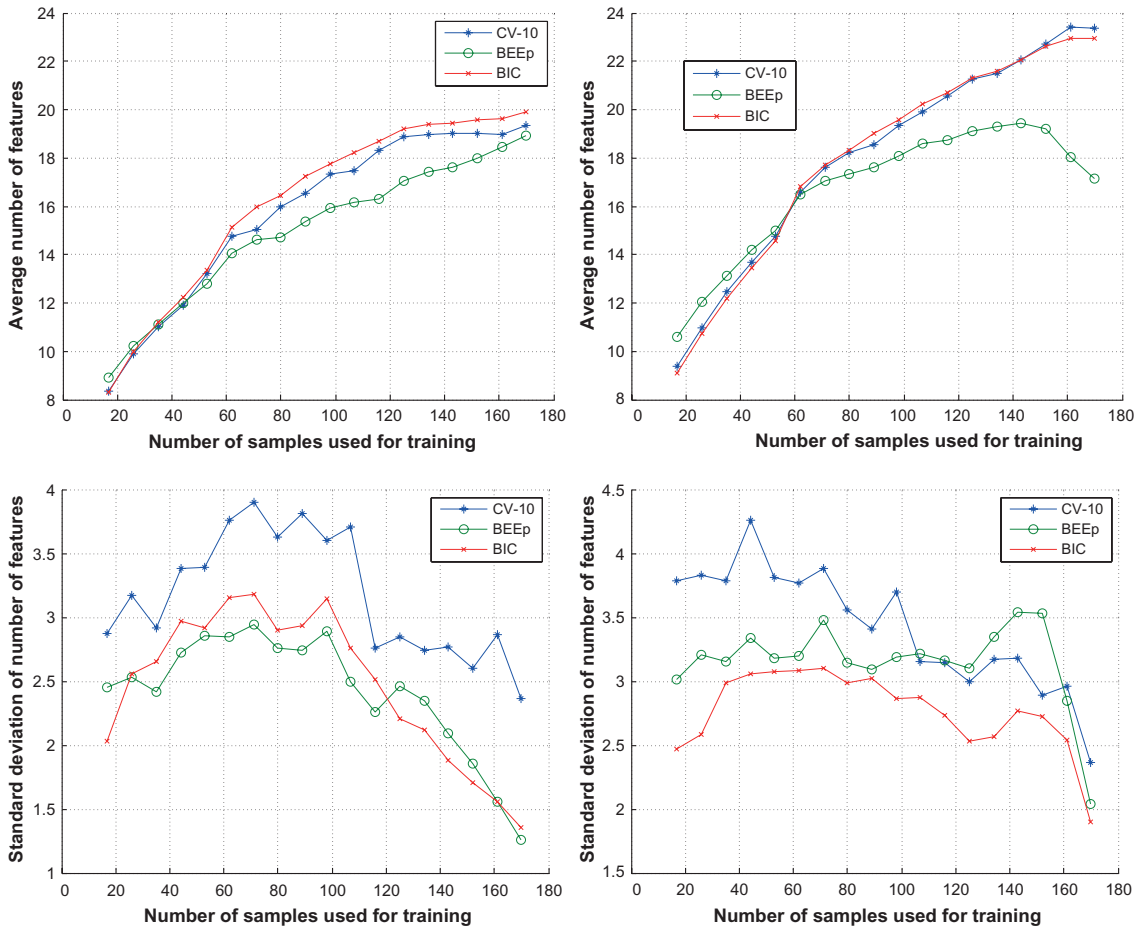


Figure 4. Comparisons of the number of selected features for CV-10, BEE with proper prior (BEEp), and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements. Top: average number of selected features. Bottom: standard deviation of number of the selected features.

average number of selected features is higher than that of others (Fig. 4 top panel).

Similarity of the selected feature sets. Another performance measurement is the stability of selecting the same feature at different iterations. For this purpose, Sørensen–Dice coefficient[25] is used, which measures the degree of similarity between selected features of two different iterations. The ranges can vary from 0 to 1. The values closest to 1 indicate a high-degree of similarity.

For different training samples, we first determine which features are selected at each iteration. As the model selection process is repeated 200 times, we estimate the similarity as the mean dice coefficient for each of the 200!/(2! × (200 − 2)!) = 19,900 possible pairs of selected feature sets. The results are shown in Figure 5. In terms of stability, the performance of BEE is substantially better than those of the other methods, as the selected feature sets are most similar with that criterion – a significant issue when trying to understand the biological mechanisms behind the data. For example, with 60% training samples, the dice coefficient of BEE is 6.03% higher than that of CV (Fig. 5 left panel). On the other hand, with 90% training samples, the dice coefficient of BEE is 5.81% higher than that of CV and 4.90% higher than that of BIC (Fig. 5 right panel). Indeed, the dice coefficient is unfavorable for CV with small training samples: The dice coefficient is lowest among the alternatives, indicating that the selected feature sets with the CV criterion have high variability.

Small sample case with a cancer cell line. For further confidence on the presented method, we analyze data from a cancer cell line in a small sample setting. As described previously, we considered the classification accuracy, AUC measure, and the number of selected variables both with and without standard deviation features (Figs. 6–8). In this case,

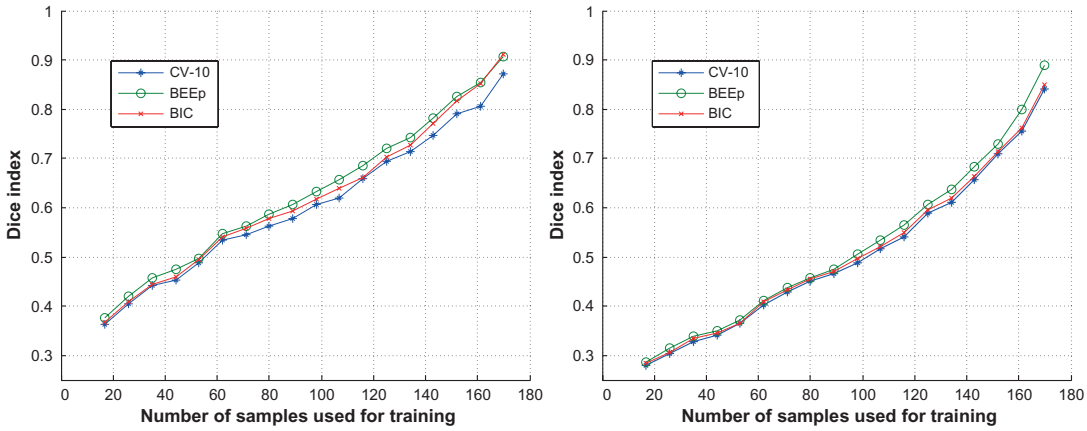


Figure 5. Comparison of the stability of selecting features for CV-10, BEEp, and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements.

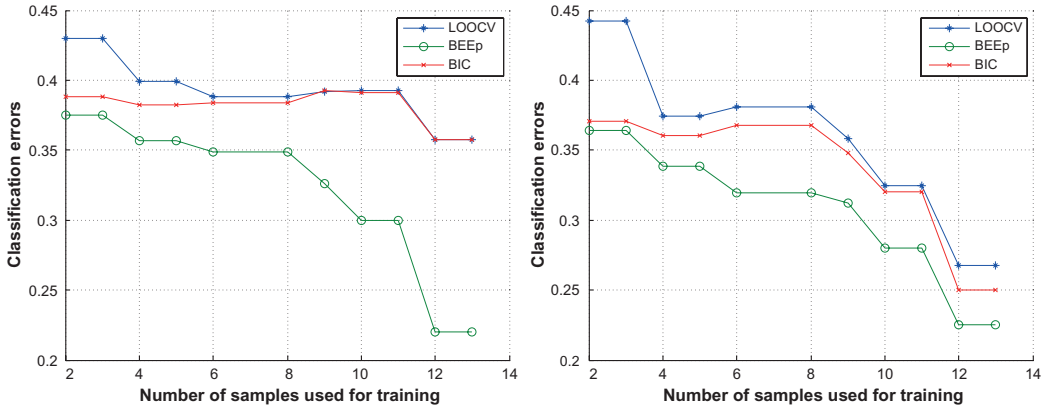


Figure 6. The average classi□cation error curves for LOOCV, BEEp, and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements.

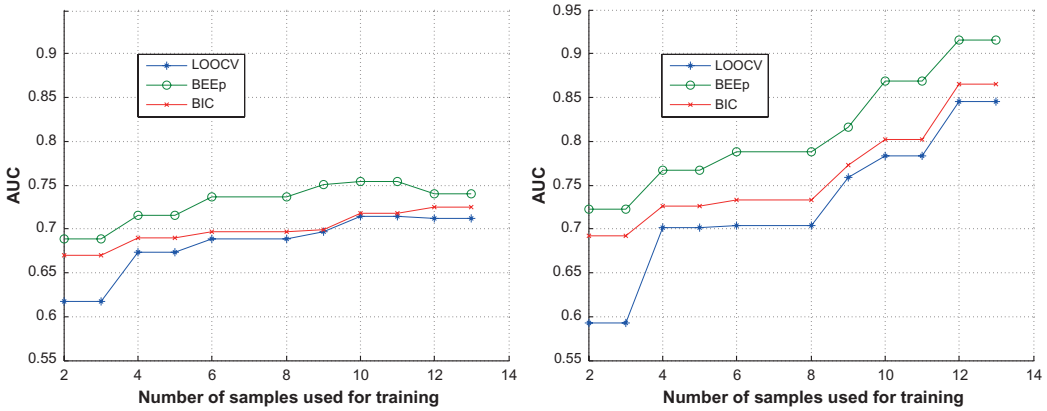


Figure 7. The average AUC curves for LOOCV, BEEp, and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements.

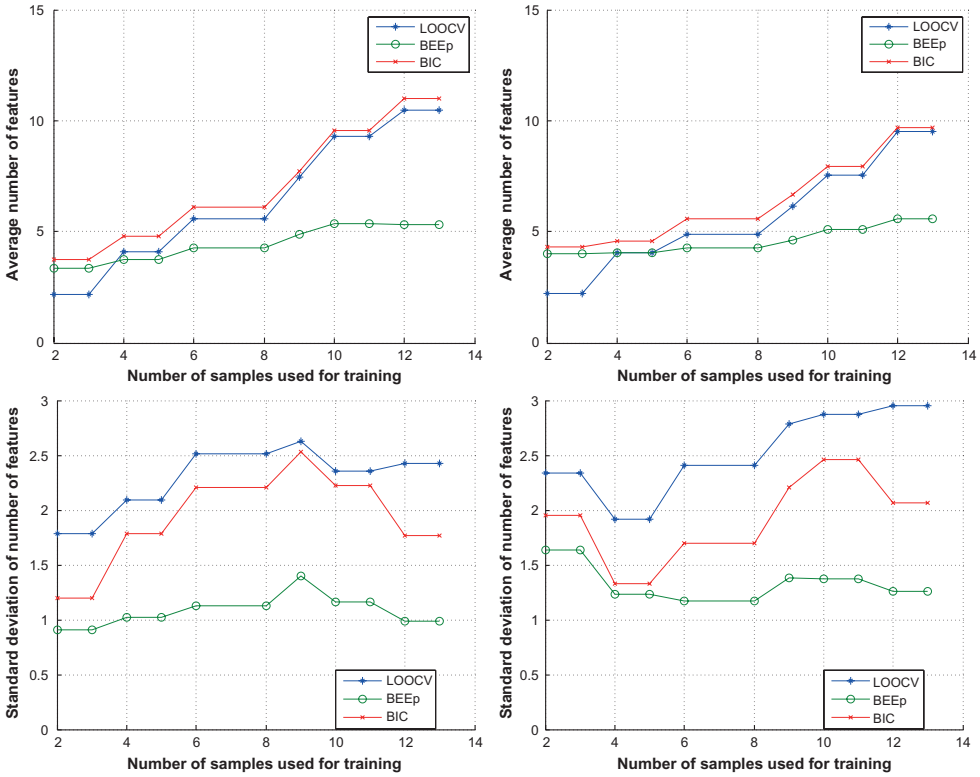


Figure 8. Comparisons of the number of selected features for LOOCV, BEEp, and BIC.
Notes: Left: feature vector of mean values of measurements. Right: feature vector of mean and standard deviation of measurements. Top: average number of selected features. Bottom: standard deviation of number of the selected features.

we split the dataset into 15%, 22%, 29%–78%, and 85% for training the classifier, while the remaining data are used for performance assessment.

As the sample size is minimal, we applied leave-one-out cross-validation (LOOCV) instead of the CV-10. The results by BEE are in general more accurate than those by BIC and LOOCV and also obtained with fewer parameters in the model. The case study with prostate cancer cell line shows that the presented method is able to efficiently classify between the treatments in a very small sample setting.

Conclusions

In this paper, we have studied the effect of feature selection classification of flow cytometry data. In particular, we considered using simplistic features instead of more complicated feature extraction pipelines widely seen in the literature. As a result, we were able to simplify and reduce the number of features without compromising the prediction accuracy. In addition to this, we considered the problem of feature selection in a small sample size setting. Such cases are not uncommon in biology, yet they have not received a lot of attention in scientific literature. In particular, the stability of the feature selection process varies a lot depending on the error measure used for model selection.

The Experimental Results section considered three different error metrics and compared them in terms of classification accuracy (measured by both classification error and AUC) and feature selection stability (measured by the number of features and the dice index between feature sets). As a result, the recently presented Bayesian error estimator (BEE) has a superior stability and an improved accuracy over the traditional counting-based approach, such as CV. The experiments show that BEE selects better classification models than the model selected by CV. In particular, the BEE is more effective compared to its alternatives when the number of training samples is relatively small.

Although in this study we concentrate only on flow cytometry data, we expect that the benefits of our approach – capability to deal with small sample settings and with high-dimensional data through reducing the number of features used for analysis – would make the method a good candidate also for other types of biomedical data. The effectiveness in model selection other types of data has already been demonstrated in Ref.[9]

Acknowledgment

We acknowledge the CSC – IT Center for Science Ltd., Finland, for the allocation of computational resources.

Author Contributions

SSH, PR, HH implemented the software and designed the experiments. LL acquired the data in the smaller dataset case. SSH wrote the manuscript. PR, HH, LL contributed to the writing and revising of the manuscript. All the authors reviewed and approved the final manuscript.

REFERENCES

1. Jennings CD, Foon KA. Recent advances in flow cytometry: application to the diagnosis of hematologic malignancy. *Blood*. 1997;90(8):2863–92.
2. Rogers WT, Moser AR, Holyst HA, et al. Cytometric fingerprinting: quantitative characterization of multivariate distributions. *Cytometry A*. 2008;73(5):430–41.
3. Bendall SC, Simonds EF, Qiu P, et al. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*. 2011;332(6030):687–96.
4. Aghaeepour N, Finak G, Consortium TF, et al. Critical assessment of automated flow cytometry data analysis techniques. *Nat Methods*. 2013;10:228–37.
5. Biehl M, Bunte K, Schneider P. Analysis of flow cytometry data by matrix relevance learning vector quantization. *PLoS One*. 2013;8(3):59401.
6. Vilar JM. Entropy of leukemia on multidimensional morphological and molecular landscapes. *Phys Rev X*. 2014;4(2):021038.
7. Manninen T, Huttunen H, Ruusuvuori P, Nykter M. Leukemia prediction using sparse logistic regression. *PLoS One*. 2013;8(8):72932.
8. Dougherty ER, Sima C, Hua J, Hanczar B, Braga-Neto UM. Performance of error estimators for classification. *Curr Bioinform*. 2010;5:53–67.
9. Huttunen H, Tohka J. Model selection for linear classifiers using Bayesian error estimation. *Pattern Recognit*. 2015;48:3739–48.
10. Kaukoniemi KM, Rauhala HE, Scaravilli M, et al. Epigenetically altered mir-193b targets cyclin d1 in prostate cancer. *Cancer Med*. 2015;4(9):1417–25.
11. Hastie T. The Elements of Statistical Learning Data Mining, Inference, and Prediction Vol. 2nd ed. New York: Springer; 2009:745.
12. Ng AY. Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*. 2004:78–85.
13. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Series B Stat Methodol*. 1996;58(1):267–88.
14. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *J R Stat Soc Series B Stat Methodol*. 2006;68(1):49–67.
15. Meier L, Van De Geer S, Bühlmann P. The group lasso for logistic regression. *J R Stat Soc Series B Stat Methodol*. 2008;70(1):53–71.
16. Candes E, Tao T. The Dantzig selector: statistical estimation when p is much larger than n. *Ann Stat*. 2007;35(6):2313–51.
17. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc Series B Stat Methodol*. 2005;67(2):301–20.
18. Friedman J, Hastie T, Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*. 2008;9(3):432–41.
19. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33(1):1–22.
20. Efron B, Gong G. A leisurely look at the bootstrap, the jackknife, and cross-validation. *Am Stat*. 1983;37(1):36–48.
21. Dalton L, Dougherty ER. Bayesian minimum mean-square error estimation for classification error – part I: definition and the Bayesian MMSE error estimator for discrete classification. *IEEE Trans Signal Process*. 2011;59(1):115–29.
22. Dalton L, Dougherty ER. Bayesian minimum mean-square error estimation for classification error – part II: the Bayesian MMSE error estimator for linear classification of Gaussian distributions. *IEEE Trans Signal Process*. 2011;59(1):130–44.
23. Huttunen H, Manninen T, Tohka J. Bayesian error estimation and model selection in sparse logistic regression. *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. Southampton: IEEE; 2013:1–6.
24. Chen J, Chen Z. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*. 2008;95(3):759–71.
25. Dice LR. Measures of the amount of ecologic association between species. *Ecology*. 1945;26(3):297–302.

Publication IV

Syeda Sakira Hassan, Rahul Mangayil, Tommi Aho, Olli Yli-Harja, and Matti Karp. "Identification of feasible pathway information for c-di-GMP binding proteins in cellulose production", *Joint Conference of the European Medical and Biological Engineering Conference (EMBEC) and the Nordic-Baltic Conference on Biomedical Engineering and Medical Physics (NBC)*, International Federation for Medical and Biological Engineering (IFMBE) Proceedings, vol 65, pp. 667-670, Jun. 2017.

Identification of feasible pathway information for c-di-GMP binding proteins in cellulose production

Syeda Sakira Hassan[1], Rahul Mangayil[2], Tommi Aho[2], Olli Yli-Harja[1] and Matti Karp[2]

[1] BioMediTech Institute and Faculty of Biomedical Sciences and Engineering, Tampere University of Technology, Tampere, Finland
[2] Laboratory of Chemistry and Bioengineering, Tampere University of Technology, Tampere, Finland

Abstract— **In this paper, we utilize a machine learning approach to identify the significant pathways for c-di-GMP signaling proteins. The dataset involves gene counts from 12 pathways and 5 essential c-di-GMP binding domains for 1024 bacterial genomes. Two novel approaches, Least absolute shrinkage and selection operator (Lasso) and Random forests, have been applied for analyzing and modeling the dataset. Both approaches show that bacterial chemotaxis is the most essential pathway for c-di-GMP encoding domains. Though popular for feature selection, the strong regularization of Lasso method fails to associate any pathway to MshE domain. Results from the analysis may help to understand and emphasis to the supporting pathways involved in bacterial cellulose production. These findings demonstrate the need for a chassis to restrict the behavior or functionality by deactivating the selective pathways in cellulose production.**

Keywords— **cyclic di-guanosine monophosphate, metabolic pathways, regularized logistic regression, random forests**

I. Introduction

The role of cyclic-di-guanosine monophosphate (c-di-GMP) as an allosteric activator for bacterial cellulose synthesis was first discovered by Benziman and coworkers [1]. Later, the group identified the genes encoding for enzymes responsible in regulating the c-di-GMP availability in *Komagataeibacter xylinus*. The synthesis and degradation of c-di-GMP are regulated by the catalytic activities of diguanylate cyclases and phosphodiesterases, respectively and identified the presence of similar domain architectures (GGDEF-EAL tandem) among them [2]. Phosphodiesterases containing either EAL or HD-GYP domains involve in c-di-GMP degradation. Genetic and biochemical evidences in several bacterial species demonstrate that the EAL domain containing proteins degrade c-di-GMP to the 5'-phosphoguanylyl-(3'-5')-guanosine (pGpG) [3]. In contrast to EAL domain, the hydrolyzing activities of c-di-GMP specific phosphodiesterases containing HD-GYP domain result in GMP rather than pGpG. However, biochemical validations are restricted due to unsuccessful purification of catalytically active HD-GYP domain containing proteins.

Besides the synthesis and degradation of c-di-GMP, proteins that function as c-di-GMP receptors are also important to elicit specific cellular function. Amikam and Galperin reported c-di-GMP binding, PilZ, in the bcsB subunit of *K. xylinus* bacterial cellulose synthase operon. Similar domain (or its homologue) was also identified in other bacterial species such as *Pseudomonas aeruginosa*, *Escherichia coli* and *Vibrio cholera*, involved in c-di-GMP mediated regulation of cellular motility, virulence and biofilm formation [4, 5]. Recently, a new c-di-GMP receptor domain, MshE, was identified in *V. cholarae* and *P. aeruginosa* that contained C-terminal ATP binding site and an N-terminal c-di-GMP binding domain [6]. Structural studies report that the c-di-GMP binding affinity of MshE domain was greater than the PilZ domain.

Taking the importance of c-di-GMP as a universal regulator for several bacterial cellular processes, with the aim to improve bacterial cellulose production, it is rational to study various targets or effectors of c-di-GMP involved in metabolic pathways. Thus, we are interested in finding significant features from the distributions of these c-di-GMP signaling pathways in diverse bacteria. We propose the use of machine learning approaches that have two advantages. First, we can identify the supporting pathways associated with c-di-GMP signaling proteins. Second, this knowledge can be applied to restrict the behavior of a new strain in synthetic biology. In order to identify relevant features, a predictive model is required that establishes the relationship between the pathways and genes encoding domains. We select in this study two state-of-the-art machine learning approaches, which yield simultaneous predictive models and feature selection.

II. Materials and Methods

A. Data

For this experiment, we considered 1024 complete bacterial genomes that are available in the NCBI's RefSeq database. The input data set is downloaded from KEGG, a

pathway database that maps for cellular and organismal functions. The output data were downloaded from [7]. The selected metabolic pathways and their significance in regulation of c-di-GMP are described in Table 1.

B. Regularized Logistic Regression

Let us consider the observations $(\mathbf{X}, y_i)$ where $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n observations and p features and $y_i \in \mathbb{R}^{n \times 1}$ in i^{th} domain with $i \in \{\text{GGDEF, EAL, HD-GYP, PilZ, MshE}\}$. A single observation x_j represents a vector of gene counts in the listed pathways and y_{ij} is the number of genes in i^{th} domain for respective bacterial genomes. A linear relationship between $\mathbf{X}$ and y_i can be modeled as $y_i = \theta_i \mathbf{X}$. Here, θ_i is the relationship parameters which can be estimated by minimizing the residual errors using the equation below.

$$\hat{\theta}_i = \arg\min_{\theta_i} \|y - X\theta_i\| \tag{1}$$

where $\|.\|$ is the standard L^2-*norm* in the parameter space [8, 9]. Although the solution is simple and easily interpretable, it is often inadequate for ill-posed behavior of the underlying data [9]. In this study, we apply a state-of-the-art regularization approach, *Least absolute shrinkage and selection operator (Lasso)*. The method provides a sparse solution by effectively shrinking the number of parameters and thereby choosing simpler model [10]. In regularization, an extra term, λ is added, which controls the trade-off between the residual error and the number of parameters. Thus, our linear model can be defined as $\hat{\theta}_i = \arg\min_{\theta_i} \|y - X\theta_i\| + \lambda \|\theta_i\|_1$, where $\lambda > 0$ is the regularization hyperparameter and $\|.\|_1$ is the L^1-*norm* in the parameter space. If we set $\lambda = 0$, it yields to Equation 1. On the other hand, a very large λ will completely shrinks the parameters to zero and may yield a null or empty model. We use the R package `glmnet` in this paper [11]. The model hyperparameter λ can be selected using the cross-validation approach [12]. In cross validation, the given dataset is randomly divided into training and testing dataset. Training dataset is used for training the model, whereas testing dataset is used for testing the model. The most common variation in cross validation is the K-fold cross validation, which is used in this study with $K = 10$.

C. Random Forests

Random forests is an ensemble learning method where collection of decision trees are built by bootstrap aggregation [13]. A decision tree can be thought of as a hierarchical representation of if-then rules, where each internal node in the tree describes each input attribute or feature and the leaf node describes the output value. The random forests combines many binary decision trees using several bootstrap samples from the data and choses randomly at each node a subset of the input features. The advantage of random forests is unbiasedness to overlearning, which is done by averaging, thereby improves the prediction accuracy. Since the algorithm ranks the importance of features, it acts as an embedded feature selection approach. In this paper, we use the R package `randomForest` developed by Liaw and Wiener [14].

D. Prediction and Feature Selection

For prediction, we train the Lasso and Random forests models with all the data set except the one for which we measure the quantity of genes encoding different domains. We use the default values of the hyperparameters.

In order to find relevant pathways, we train the models with the complete data set and estimate the importance of the pathways listed in Table 1. The regularization technique embedded in Lasso allows removing the irrelevant features from the dataset by setting the coefficient values to zero. For random forest, we use the function `importance()` to rank the features based on out-of-bag prediction error. Larger values signifies the importance of the features.

III. Results and Discussion

A. Results

In our case study, first we demonstrate the prediction performance of different methods for representative bacterial genomes. Then, we investigate the feature selection approach to find significant pathways associated with c-di-GMP binding proteins in bacterial cellulose production. Figure 1 illustrates the true and predictive distribution of genes encoding GGDEF domain for representative bacterial genomes. Here, GGDEF domain is presented as an example. Similar distributions can be drawn for other domains.

In feature selection, the coefficient values obtained by Lasso and random forests models can be visualized through the heatmap representation in Figure 2. The higher the coefficient values, the more significant the pathways associated with c-di-GMP signaling domains. In comparison to the metabolic pathways evaluated using Lasso and random forests methods, a denser heatmap is observed for bacterial chemotaxis pathway, indicating its significance for c-di-GMP encoding domains. The lipopolysaccharides biosynthesis is also considered relevant by the methods. The results are analogous to our hypotheses listed in Table 1.

Table 1: List of metabolic pathways and their impact on bacterial growth.

Metabolic pathways	Significance
Glycolysis / Gluconeogenesis	Metabolizes glucose to pyruvate.
Citrate cycle (TCA cycle)	Generic pathway involving in ATP/GTP production, where c-di-GMP acts as the precursor.
Pentose phosphate pathway	Two stage (oxidative and non-oxidative) anabolic pathway generating RNA and aminoacid precursors from glucose.
Starch and sucrose metabolism	Pathway linking the metabolism of complex carbon substrates such as starch and sucrose to glycolysis route.
Amino sugar and nucleotide sugar metabolism	Pathway involves degradation of amino and nucleotide sugars producing sugar derivatives through glycosylation reaction.
Lipopolysaccharide biosynthesis	C-di-GMP positively regulates this pathway. The pathway is linked with nucleotide sugar metabolism and pentose phosphate pathway.
Sphingolipid metabolism	Pathway involving in the breakdown of lipids that contain sphingoid backbone bases to ceramides. Not commonly present in bacteria.
Terpenoid backbone biosynthesis	Pathway initiates with the condensation of glyceraldehyde 3-phosphate and pyruvate from glycolysis route to produce isoprenoids.
Biosynthesis of amino acids	This pathway is required for cell growth.
ABC transporters	Multi-subunit protein family involved in the import (nutrients, trace metals, and vitamins) and export (metabolites) within the bacterial cell.
Bacterial chemotaxis	Movement of bacteria in response to chemical stimulus. This pathway is regulated by c-di-GMP levels.
Phosphotransferase system (PTS)	Active transport of extracellular substrates into the bacterial cell.

B. Discussion

We examined the feature selection approaches of Lasso and random forests algorithms for c-di-GMP binding proteins. For the first four domains (GGDEF, EAL, HD-GYP, PilZ), the performance of both methods is almost equivalent without requiring any fine-tune for the model hyperparameters. For MshE domain, the random forests method has shown to be competitive with the alternative Lasso approach. According to Lasso, none of the selected pathways is significant for MshE domain. One possible reason is that, the number of genes encoding MshE domain is few. Therefore, the lasso may fail to associate the significance between the pathways and the MshE domain. On the other hand, the nonlinearity inherent in random forests approach facilitates to identify the significance of the pathways.

IV. Conclusion

The prominent directions in c-di-GMP research has yielded more questions than answers. The addition of machine learning approaches can offer a new insight and understanding in regulation of c-di-GMP binding proteins. We demonstrate in this study only 12 pathways from KEGG database which is updated constantly. With our approach, it is also possible to integrate and interpret large-scale data set from KEGG database. Identifying relevant metabolic pathways can be an attractive strategy, for example in synthetic biology, by which we can inactivate or silent activate cryptic pathways in bacterial strain for advancement in cellulose production.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

1. Ross P., Weinhouse H., Aloni Y., et al. Regulation of cellulose synthesis in *Acetobacter xylinum* by cyclic diguanylic acid *Nature.* 1987;325:279-281.
2. Tal R., Wong H. C., Calhoon R., et al. Three cdg operons control cellular turnover of cyclic di-GMP in *Acetobacter xylinum*: genetic organization and occurrence of conserved domains in isoenzymes *Journal of Bacteriology.* 1998;180:4416-4425.

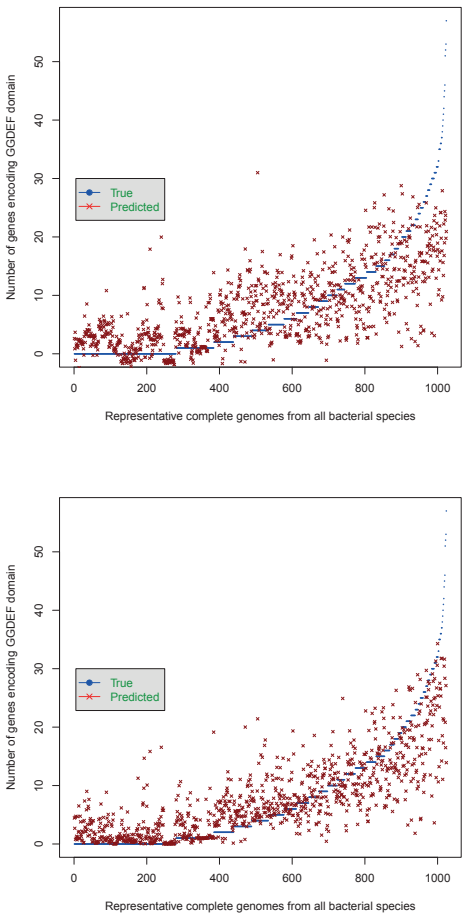


Fig. 1: Distribution of genes encoding c-di-GMP signaling GGDEF domain in respective complete genomes across all bacterial species. Blue circle represents the true distribution and red circle represents the predicted distribution using Lasso (top panel) and random forests (bottom panel) models.

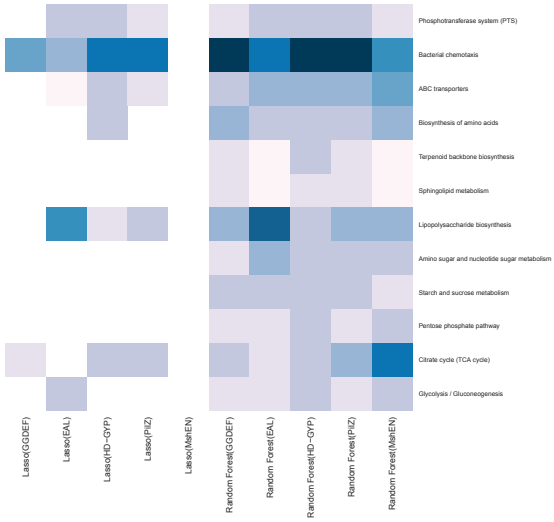


Fig. 2: Significance of pathways represented by heatmap.

3. Bobrov Alexander G., Kirillina Olga, Perry Robert D.. The phosphodiesterase activity of the HmsP EAL domain is required for negative regulation of biofilm formation in *Yersinia pestis FEMS microbiology letters.* 2005;247:123-130. pmid:15935569.
4. Pratt Jason T., Tamayo Rita, Tischler Anna D., Camilli Andrew. PilZ domain proteins bind cyclic diguanylate and regulate diverse processes in *Vibrio cholerae Journal of Biological Chemistry.* 2007;282:12860-12870. pmid:17307739.
5. Alm Richard A., Bodero Amanda J., Free Patricia D., Mattick John S.. Identification of a novel gene, pilZ, essential for type 4 fimbrial biogenesis in *Pseudomonas aeruginosa Journal of Bacteriology.* 1996;178:46-53. pmid:8550441.
6. Jones Christopher J., Utada Andrew, Davis Kimberly R., et al. C-di-GMP regulates motile to sessile transition by modulating MshA pili biogenesis and near-surface motility behavior in *Vibrio cholerae PLoS Pathog.* 2015;11:e1005068.
7. Römling U., Galperin M. Y., Gomelsky M.. Cyclic di-GMP: the first 25 years of a universal bacterial second messenger *Microbiology and molecular biology reviews : MMBR.* 2013;77:1-52.
8. Stigler Stephen M. Mathematical statistics in the early states *The Annals of Statistics.* 1978;6:239–265.
9. Hastie Trevor, Tibshirani Robert, Friedman Jerome. *The elements of statistical learning: data mining, inference, and prediction*;2nd edition. New York: Springer 2009.
10. Tibshirani Robert. Regression shrinkage and selection via the lasso *Journal of the Royal Statistical Society Series B (Methodological).* 1996;58:267-288.
11. Friedman Jerome, Hastie Trevor, Tibshirani Robert. Regularization Paths for Generalized Linear Models via Coordinate Descent *Journal of Statistical Software.* 2010;33:1–22.
12. Efron Bradley, Gong Gail. A leisurely look at the bootstrap, the jackknife, and cross-validation *The American Statistician.* 1983;37:36-48.
13. Breiman Leo. Random forests *Machine Learning.* 2001;45:5-32.
14. Liaw Andy, Wiener Matthew. Classification and Regression by randomForest *R News.* 2002;2:18-22.

Author: Syeda Sakira Hassan
Institute: Tampere University of Technology
Street: Korkeakoulunkatu 10
City: Tampere
Country: Finland
Email: sakira.hassan@tut.fi

Publication V

Syeda Sakira Hassan, Jari A. Niemi, Jussi Tohka, and Heikki Huttunen, "Bayesian Receiver Operating Characteristic Metric for Linear Classifiers", *Pattern Recognition Letters, in review.*